NONLINEAR STRUCTURAL ANALYSIS OF AN ICOSAHEDRON AND ITS APPLICATION TO LIGHTER THAN AIR VEHICLES UNDER A VACUUM

THESIS

Ruben Adorno-Rodriguez, Second Lieutenant, USAF

AFIT-ENY-14-M-03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-14-M-03

NONLINEAR STRUCTURAL ANALYSIS OF AN ICOSAHEDRON AND ITS APPLICATION TO LIGHTER THAN AIR VEHICLES UNDER A VACUUM

THESIS

Presented to the Faculty

Department of Aeronautical and Astronautical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Ruben Adorno-Rodriguez, BSME

Second Lieutenant, USAF

March 2014

AFIT-ENY-14-M-03

NONLINEAR STRUCTURAL ANALYSIS OF AN ICOSAHEDRON AND ITS APPLICATION TO LIGHTER THAN AIR VEHICLES UNDER A VACUUM

Ruben Adorno-Rodriguez, BSME
Second Lieutenant, USAF

Approved:

//signed//	5 Mar 2014
Anthony Palazotto, PhD (Chairman)	Date
//signed//	10 Mar 2014
William Baker, PhD (Member)	Date
//signed//	5 Mar 2014
Lt. Col. Anthony M. Deluca, PhD (Member)	Date

Abstract

The concept that a structure is capable of producing buoyancy using an internal vacuum rather than a gas dates back to the 1600s; but material technology has restricted the construction of such concept for common geometries, such as the sphere. Different and often complex geometries compensate for the lack of light materials that provide the stiffness and strength needed. Therefore, this research looks at an Lighter than Air Vehicle (LTAV) in the form of an icosahedral frame/skin configuration using nonlinear finite element analysis in order to determine the structural response of such vehicle, its capacity to sustain a vacuum with both material technology that exists today and in the near future, and its buoyancy characteristics. The structural response is characterized with large displacements; where membrane behavior dominates the icosahedral skin response, generating geometric stiffening in the overall structure. It is shown that those displacements have minimal effect in the structures buoyancy, with no more than 4% reduction. Overall, the nonlinear analysis of the icosahedral structure provided tangible background on its behavior and the Lighter than Air Vehicle (LTAV) applicability. It is feasibly possible to actually manufacture this type of vehicle in the very near future depending upon newer materials with more advanced strength.

I dedicate my thesis work to my beloved wife and parents, who have always been there for me, especially to my wife for her unconditional love and support.

Acknowledgments

I would like to express my deepest appreciation to my advisor, Dr. Anthony N. Palazotto, for his continuous and tireless support and guidance throughout the course of this research. His undeniable dedication, support, knowledge and motivation lifted my spirit through this process. I would like to thFank Trent T. Metlen and Brian Cranston for your contributions and commitment to this research. I would also like express my gratitude to Dr. William Baker for taking out is time to provide a new perspective and creativity to this research. Last but not least, I would like to express my appreciation to Dr. David Stargel from the Air Force Office of Scientific Research for his interest in this research and sponsorship.

Ruben Adorno-Rodriguez

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

List of Symbols

Symbol	Definition
a_c	Circular Membrane Radius
a_s	Half the Edge Length of the Square Membrane
A	Area
B	Buoyancy
c	Beam Thickness to Radius Ratio
E	Modulus of Elasticity
h	Height of the Equilateral Triangle
H	Altitude
I	Area Moment of Inertia
k	Distributed Load
K	Linear Stiffness Matrix
K_T	Tangent Stiffness Matrix
l_{beam}	Icosahedron Edge Length
L	Column Length
g	Acceleration of Gravity
M	Mass
$M_{air,i}$	Internal Air Mass
$M_{air,o}$	External Air Mass
M_b	Bending Moment
M_{frame}	Frame Mass

Symbol	Definition
M_{skin}	Skin Mass
M	Mass
N_1	Vector - Linear Function of q
N_2	Vector - Quadratic Function of q
N_x	Membrane Force in the x Direction
N_y	Membrane Force in the y Direction
N_{xy}	Membrane Force in the xy Direction
P	Pressure
$P_{air,i}$	Internal Air Pressure
$P_{air,o}$	External Air Pressure
$P_{applied}$	Air Pressure Applied to or 'Felt' by the Skin
P_{cr}	Critical Load
P_o	Sea Level Pressure
q	Nodal Displacements Vector
r	Icosahedron Radius
r_{beam}	Beam Radius
r_c	Circular Membrane Radial Position
R	Nodal Loading Vector
Rs	Specific Gas Constant
S	Stiffness
SF	Safety Factor
S_f	Design Failure Strength

Symbol	Definition
S_o	Stress at the Center
S_u	Ultimate Strength
S_y	Yield Strength
t	Membrane Thickness
t_{beam}	Beam Thickness
t_{skin}	Icosahedral Skin Thickness
$T_{air,i}$	Internal Air Temperature
$T_{air,o}$	External Air Temperature
T_o	Temperature at Sea Level
u	Displacement in the x Direction
v	Displacement in the y Direction
V	Volume
V_{frame}	Frame Volume
V_i	Icosahedron Internal Volume before Deformation
V_r	Icosahedron Internal Volume Reduction
V_s	Strain Energy
V_{skin}	Skin Volume
w	Vertical Displacement
w_o	Center Displacement
W	Weight
y	Beam Distance from the Neutral Axis

Symbol	Definition
Greek Symbols	
δ	Maximum Displacement
ε_x	Strain in the x Direction
ε_y	Strain in the y Direction
ε_{xy}	Strain in the xy Direction
ν	Poisson's Ratio
ρ_{air}	Air Density at Seal Level
$\rho_{air,i}$	Internal Air Density
$\rho_{air,o}$	External Air Density
ρ_{frame}	Frame Density
ρ_{skin}	Skin Density
σ_b	Bending Stress
σ_f	Design Failure Stress
σ_n	Stress in the Directions $n = 1, 2$ *and* 3

List of Acronyms

Acronym	Definition
BC	Boundary Condition(s)
CAE	Complete Abaqus Environment
CNT	Carbon Nanotubes
DOF	Degrees of Freedom
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Methods
LTA	Lighter than Air
LTAV	Lighter than Air Vehicle
SL	Sea Level
W/B	Weight to Buoyancy Ratio

NONLINEAR STRUCTURAL ANALYSIS OF AN ICOSAHEDRON AND ITS APPLICATION TO LIGHTER THAN AIR VEHICLES UNDER A VACUUM

I. Introduction

1.1 Objective

Aircraft structures have been designed for more than a century with wing like configurations; tremendous progress has been made in this direction. The research presented in this thesis is an attempt to evaluate a different type of air structure: a structure that relies on the effect of buoyancy through an internal vacuum to provide lift rather than the normal wing. Therefore, the objective is to evaluate the characteristics of LTAV subjected to a vacuum, pointing out the structural features for consideration in the eventual design of such a vehicle.

In order to evaluate the vacuum LTAV, Archimedes principle with the ideal gas law along with nonlinear Finite Element Analysis (FEA) with the Newton Raphson technique are used. The Archimedes principle states that an object submerged in any fluid exerts a buoyant force equal to the weight of the displaced fluid, establishing the relationship that allows the design to become Lighter than Air (LTA). Furthermore, the ideal gas law serves to express the air density in terms of pressure and altitude, proving a direct venue between the atmospheric pressure and the pressure that the structure is subjected to. FEA then provides the means to evaluate the nonlinear behavior of the structure and its relationship to buoyancy.

1.2 Chronology of Lighter than Air Vehicles

The idea of having a structure, or a vehicle for that matter, float in air dates back to the 1600s, when Italian monk Francesco Lana de Terzi proposed the 'Aerial Ship'. He wrote:

> The preceding inventions did not exhaust the ardour or the curiosity of the human intellect, but have, rather, spurred it to seek how men could, after the fashion of the birds, also fly in the air... No one has, however, deemed it possible so to construct a vessel that it would travel on the air as if it were supported on water, insomuch that it has not been thought practicable to make a machine lighter than the air itself, which it is necessary first to do in order to accomplish the desired end [28, 11].

The main idea behind this statement is the concept of having an object be LTA. He proposed an LTAV composed of four LTA evacuated spheres made of copper supporting a basket, as shown in Figure 1. This design brought various objections, made by scientists of that time and later answered by Lana, in which all were proven wrong but one. Those objections included the issue of evacuating the air out of the spheres, the 'unstoppable' rising of the vehicle once afloat, and the capacity of the spheres to remain rigid after evacuation [26]. First, evacuating, or

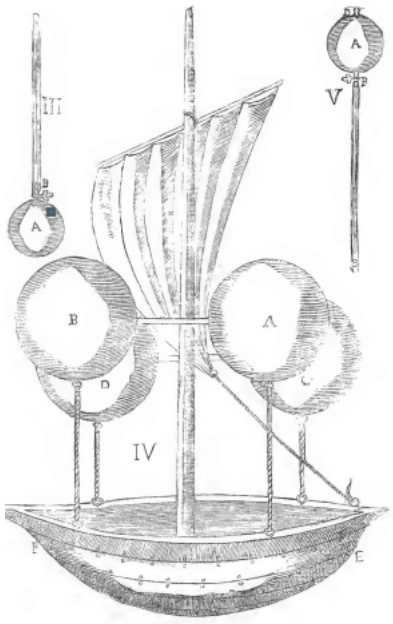


Figure 1: Aerial Ship: lighter that air ship design proposed by Francesco Lana de Terzi in 1670 [28, 15]

creating a vacuum, inside any enclosed structure presents no challenge nowadays, and back then it could be accomplished with Boyle's air pump. Second, departing from the fact that the air density reduces with altitude, the vehicle or structure will stop rising once the weight of the displaced air equals the weight of structure itself. The last objection, which challenges the capacity of the spheres to remain rigid, is still valid today. After all, realizing that a structure could float in air was the most important finding, and underlying basis for the LTAV concept.

In 1709, the first LTAV design in the form of a hot air balloon was launched by the Montgolfier brothers [14]. The hot air balloon uses a heat source to reduce the density of air inside, creating a pressure difference that produces buoyancy. This understanding led to the creation of dirigibles. First introduced by Henri Giffard in 1852 [10], dirigibles rely on the use of a gas inside the structure that is LTA, commonly Helium or Hydrogen, to displace enough air volume such that the weight of the structure and the internal gas is less than the displaced air itself, acquiring buoyancy. Dirigibles became the first air vehicle, capable of traveling and maneuvering with the assistance of propeller and control surfaces. A pictographic representation of the history of dirigibles development from 1850 to 1960 is shown inFigure 2. Three major design types arose: non-rigid, semi-rigid and rigid. A rigid dirigible is one that uses a framework to retain its shape rather being forced into shape by the internal gas, as with the non-rigid. The semi-rigid contains a partial framework mainly used to distribute suspension and lifting loads.

Non-rigid designs have historical significance because they mark the start of the dirigible era, given by the Giffard, shown at the lower left of Figure 2. The Giffard was 44 m (144 ft 4 in) long and hydrogen filled. In the need of carrying heavy payloads, rigid

designs became popular, making them large compared to non-rigids and semi-rigids. The primary rigid design was developed by the Zeppelin Company of Germany, and in fact, the term 'Zeppelin' became representative of all rigid designs, most made by this company. They were mainly recognized for their use in commercial transportation, such as the famous Hindenburg (LZ-129), but some were used for military purposes, such as ZR-1 used by the United States Navy [20]. The LZ-129 was 803.8 ft (245 m) long and hydrogen filled.

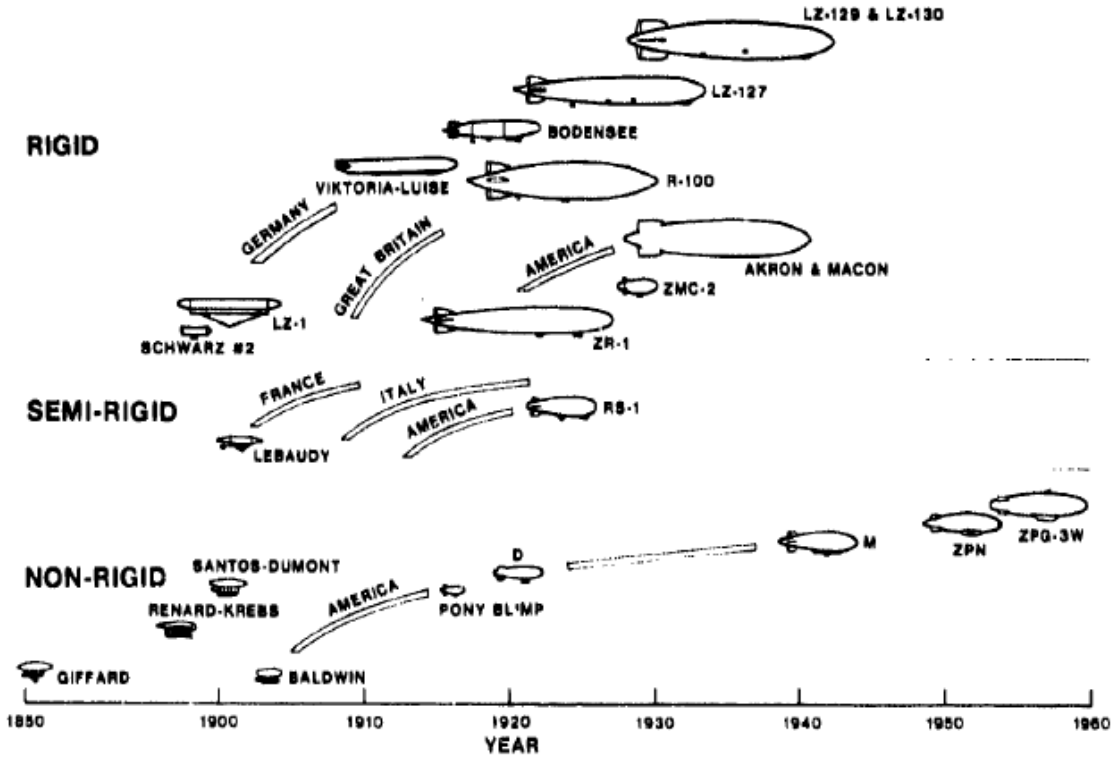


Figure 2: History of Dirigibles Development from 1850 to 1960, including rigid, semi-rigid and non-rigid designs [3]

A historical question arises: 'Why, after so many years of development, did the dirigibles vanish (for the most part)?' Dirigibles had major challenges over those years, including their speed and control limitations, safety and poor ground handling. The advancements of heavier than air vehicles, i.e., airplanes, during that period, such as their high speed and control capacity, was the main reason for the disappearance of dirigibles. Furthermore, airplanes eliminated most of the safety and handling issues. Safety mostly relates through history with the use of hydrogen as a lifting gas. Being flammable, hydrogen has been reported to be the cause of more than 22 accidents related to dirigibles from 1930 to 1937 [21]. Helium, on the other hand, is a inert gas and therefore it has been used since 1960s in dirigibles, but it being a depleting nonrenewable energy source, has created restraint in LTAV designs throughout the years.

In the last decade, technology has driven new, safer and efficient LTAV designs. The Lockheed P-791 hybrid air vehicle, having its first flight on January, 2006 [16], is an example of these designs. Hybrid designs take advantage of aerodynamics in combination to its buoyancy to produce lift and movement. These designs have solved most of the safety and ground handling issues that previous dirigibles had, but they still rely on Helium as the lifting gas.

In the same way, today's advantages in materials and manufacturing techniques makes producing buoyancy by evacuating a structure (creating a vacuum inside) an idea that is not as far-fetched as when Lana suggested it. He suggested the use of vacuum spheres. The sphere is the ideal shape for a vacuum LTAV since it achieves the greatest stiffness with the minimum weight, therefore maximizing buoyancy; but a material that has enough specific stiffness, $E^{1/2}/\rho$, where E is the *modulus of elasticity* and ρ is the

density[1], such that an homogeneous sphere can resist a vacuum has yet to be created or found for that matter. Therefore, designers have resorted to other geometries to compensate the lack of material stiffness. Three of the LTAV designs found in published literature that use an internal vacuum are mentioned here. First, A. Akhmeteli and A.V. Gavrilin proposed the use of layered shell spheres, a sandwich construction type, and second, T.T. Metlen considers the icosahedron and rotating cylinders. Details are discussed in Section 2.8.

From the structural point of view of vacuum LTAV, a rigid design is needed since there is no internal gas to force it into shape. Imagine a simple balloon: inflating it with helium would cause it to float; since the air displaced weighs more than the balloon and helium themselves. However, if the helium is vacated out of the balloon, the balloon would shrink and no internal volume will be left, such that the balloon becomes heavier than air. On the other hand, a rigid structure, or a rigid balloon for this matter, can maintain its internal volume once vacated, provided it is stiff enough to resist the external forces.

The research presented in this thesis tries to answer questions that arise from Metlen's research by evaluating the icosahedral structure with nonlinear analysis. Therefore, the geometric characteristics of the icosahedron are presented next to provide background on the reasoning of selecting such a geometry.

[1]Specific stiffness ($E^{1/2}/\rho$) is a material index that establishes the relative material performance. In this case, $E^{1/2}/\rho$ was used to minimize weight while maximizing stiffness. Derivation and details of this index are discussed in Section 2.7.

1.3 The Icosahedron as a Geometrical Shape

The icosahedron, properly called a regular icosahedron, is a regular polyhedron and platonic solid [51]. The word 'regular' refers to polygons that are characterized by sides of the same size, located symmetrically about a common center [53], such as the equilateral triangle and the square. A polyhedron is then regular if its faces and vertex figures are regular [54]. Furthermore, a platonic solid is a convex polyhedron (that can be algebraically defined as the set of solutions to a system of linear inequalities) with equivalent faces composed of regular polygons [52].

The icosahedron has several advantages that revolve around one characteristic: symmetry. Symmetry results from the 20 equilateral triangles that form the icosahedron. The icosahedron and its decomposition into 20 triangles is shown in Figure 3. As a symmetry byproduct, a circumscribed sphere touches each of the 12 vertices that make the icosahedron, such that an icosahedral radius is defined as the distance from the center to each vertex. From the structural point of view, symmetry provides many advantages including uniform stress distribution, simplified construction (compare to other polyhedrons) and modeling simplifications. The latter becomes important since having a simplified structure can yield an accurate model. One of the modeling simplifications is the use of one triangle to approximate the behavior of the structure, done with the triangle submodel (discussed in Section 3.5).

1.4 Challenges of Vacuum Lighter than Air Vehicle

Stiffened structures, such as monocoque structures use in airplanes, sandwich structures used in panels and geodesic structures used in domes, provide stiffened

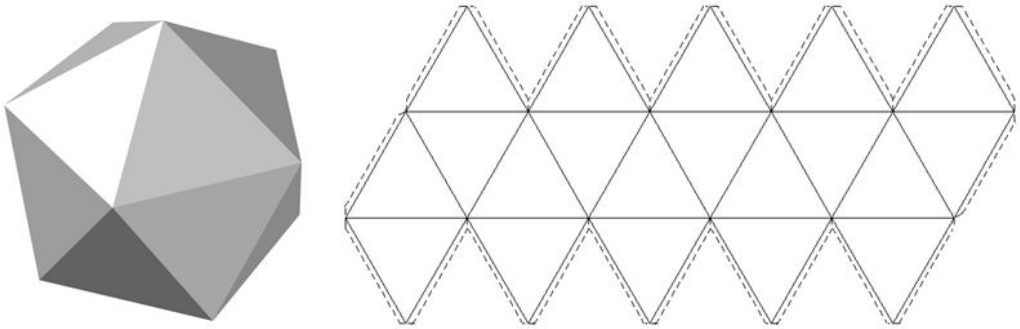


Figure 3: Icosahedron, left: three dimensional shape [9], right: planar decomposition [18]

alternatives to homogeneous structures that minimize weight while increasing the critical load[2]. Furthermore, new manufacturing technologies, materials and analysis techniques, such as Finite Element Methods (FEM), allows us to consider far more complicated structures, opening the door to new designs such as the sandwich type structure and the icosahedron. These structures usually exhibit shell, membrane and beam like behavior[3].

An internal vacuum applied to shell and beam like structures introduces various design challenges, to include structural instability and integrity. Structural instability is a byproduct of buckling, in the forms of bifurcation and collapse, the last an inherently nonlinear problem. On the other hand, structural integrity relates to the structure's capacity of withstanding the applied load without material failure and in this case for the deflected structure to maintain enough internal volume such that it is still buoyant. Both

[2]The critical load is defined as the load that produces buckling of the structure. See Section 2.6 for more information.

[3]Shell and membranes are defined as structural parts that are initially curved and flat, respectively, where their thickness is much smaller than their characteristic length [49, 1]. See Section 2.5 for more information. A beam is defined as a structural slender part subjected to transverse loading.

structural stability and integrity have a direct relationship with the geometric shape and material of the structure, and can cause failure.

FEM are used in this thesis to model the behavior of an icosahedral skin reinforced by an icosahedral frame under a vacuum. Models use nonlinear analysis to evaluate the stability and integrity of the structure, including the behavior of the skin with respect to its thickness and the frame response to the applied vacuum. Different materials and the structure's capacity to achieve buoyancy are also considered.

1.5 Assumptions and Limitations

FEA inherently includes assumptions and approximations, starting with the discretization of a complex structure such as the icosahedron. Nonetheless, the following assumptions where made within the FEA realm:

1. Skin Submodel[4]
    (a) The frame remains rigid as pressure is applied.
    (b) The skin acts like a membrane such that simply supported Boundary Condition(s) (BC) along the triangle edges are used[5].
2. Frame Submodel[6]
    (a) All the load applied to the skin is distributed directly to the frame with no moments.

[4]Model details in Section 3.5

[5]A convergence study is conducted using both membrane and shell elements against analytical solutions. See Chapter 3.

[6]Model details in Section 3.6

(b) Frame members act like beams; beam elements are therefore used for analysis.

3. Icosahedron Model

   (a) The skin behaves as a membrane such that only the displacement Degrees of Freedom (DOF) are tied to the frame members.

   (b) The skin edges displace along with the frame edges, therefore frame and skin edges share nodes.

4. All materials are modeled as linear elastic. Finding realistic material properties that allow the icosahedron to achieve positive buoyancy is critical, therefore having material properties tied to material names while making the isotropic assumption provides perspective while maintaining simplicity.

5. The air behaves as an ideal gas, limiting the maximum altitude at which the buoyancy equations are valid to 65,000 ft.

1.6 Overview

- Chapter I: States the objective of this thesis, the chronology of LTAV, the icosahedron as a geometrical shape and the challenges of an LTAV.

- Chapter II: Summarizes theory presented in relevant literature related to the structural behavior and failure modes of shell and beam like structures along with the buoyancy relationships of the icosahedron.

- Chapter III: Presents the development of the different models used in this thesis, to include the modeling processes, the Finite Element (FE) techniques and the convergence studies.
- Chapter IV: Presents the icosahedron models' results and comparison for different material properties.
- Chapter V: Summarizes the research, and presents conclusions drawn, their significance, and recommendations for future research.
- Appendix A: Includes the tabulated results of the different studies made in order to develop the icosahedron model.
- Appendix B: Includes the Python codes used to create, analyze and extract results from the different models considered.
- Appendix C: Includes all the Matlab codes used to run the icosahedron models and extract their results.

II. Theory

2.1 Overview

The effect of this research expands two main areas: the understanding of the icosahedron's structural behavior and its applicability to LTAV subjected to a vacuum. The field of structures subjected to an internal vacuum has limited published literature, with no literature found on the structural instability of an icosahedron subjected to a vacuum. Furthermore, only two published researches were found related to LTAV subjected to an internal vacuum. On the other hand, individual structural components have been extensively studied.

The purpose of this chapter is three-folded: (1) to state the principles of LTAV; (2) to summarize the research that has been done on vacuum LTAV and subjects related to the behavior of the icosahedron's structural components; and (3) to state its relationship to the research presented in this thesis. The discussion starts with the Weight to Buoyancy Ratio (W/B) concept, its applicability to the icosahedron, and the effect of altitude on structural loading and W/B. Then it moves to a discussion of nonlinear analysis, followed by the structural instability, failure and the behavior of membranes and shells. Then, relevant materials and their properties are considered; finalizing with the summary of vacuum LTAV research.

2.2 Weight to Buoyancy Ratio

"Any object, wholly or partially immersed in a fluid, is buoyed up by a force equal to the weight of the fluid displaced by the object.", Archimedes of Syracuse [22]. In other

words, Archimedes principle states that when a body is submerged in a fluid, a vertical (buoyant) force equal to the weight of fluid displaced by set body is produced. In a fluid column, pressure increases as altitude decreases, caused by the weight of the overlying fluid. Thus, a submerged body experiences a greater pressure at the bottom of the column than at the top. If the weight of the immersed body is more than the weight of the fluid it is displacing, the body will tend to sink. On the other hand, if the weight of set body is less than the weight of the fluid it is displacing, the body will tend to float. The point at which both weights equal is the point of neutral buoyancy, where the body remain static provided no other force is exerted on or by the body. Ergo, in order for a body to be buoyant, its weight has to be less that the weight of the fluid displaced by it. This relationship among body and fluid weights results in the W/B concept.

The W/B is a concept that establishes how buoyant a structure is with respect to its own weight. In ideal conditions, this ratio tends to zero, such that its weight is much less than its buoyant force, producing lift. In case of the icosahedron subjected to a vacuum, we have two main components, the frame and the skin, as shown in Figure 4. The icosahedron W/B is then given by:

$$\frac{W}{B} = \frac{\left(M_{skin} + M_{frame} + M_{air,i}\right) g}{M_{air,o} g} = \frac{V_{skin}\,\rho_{skin} + V_{frame}\,\rho_{frame} + (V_i - V_r)\,\rho_{air,i}}{(V_i - V_r)\,\rho_{air,o}} \tag{2.1}$$

where:

B = buoyancy of the structure

g = acceleration of gravity

$M_{air,i},\ M_{air,o}$ = internal and external air masses, respectively

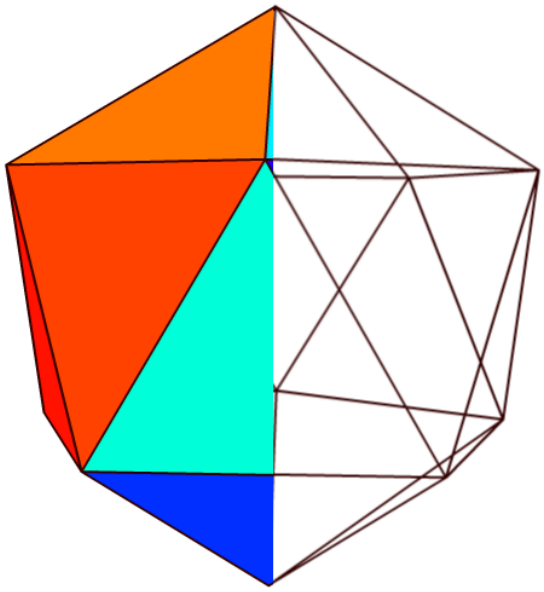


Figure 4: Icosahedral Frame/Skin Combination - skin (left half), frame(right half) [31]

M_{frame}, M_{skin} = frame and skin masses, respectively

V_{frame}, V_{skin} = frame and skin volumes, respectively

V_i = icosahedron internal volume before deformation

V_r = icosahedron internal volume reduction

W = weight of the structure

$\rho_{air,i}$, $\rho_{air,o}$ = internal and external air densities, respectively

ρ_{frame}, ρ_{skin} = frame and skin densities, respectively

The internal volume reduction accounts for the deflection of the loaded skin and/or any internal component. The internal and external air densities, $\rho_{air,i}$ and $\rho_{air,o}$, are variables that depend on altitude and amount of vacuum applied. To express the air densities in terms of pressure, the air can be modeled as an ideal gas using the ideal gas law given by:

$$P = \rho RT \tag{2.2}$$

where:

P = pressure

ρ = air density

Rs = air specific gas constant

Using the ideal gas law to express the densities in terms of pressure and temperature, provides the means to relate the buoyancy of the icosahedron with altitude and vacuum level, extending the design envelope of these equations. Substituting Equation (2.2) into Equation (2.1) results in:

$$\frac{W}{B} = \frac{V_{skin}\,\rho_{skin} + V_{frame}\,\rho_{frame} + (V_i - V_r)\left(\frac{P_{air,i}}{RT_{air,i}}\right)}{(V_i - V_r)\left(\frac{P_{air,o}}{RT_{air,o}}\right)} \tag{2.3a}$$

$$\frac{W}{B} = \frac{V_{skin}\,\rho_{skin} + V_{frame}\,\rho_{frame}}{(V_i - V_r)\left(\frac{P_{air,o}}{RT_{air,o}}\right)} + \frac{P_{air,i}}{P_{air,o}}\frac{T_{air,o}}{T_{air,i}} \tag{2.3b}$$

where:

$P_{air,i},\ P_{air,o}$ = internal and external air pressures, respectively

$T_{air,i},\ T_{air,o}$ = internal and external air temperatures, respectively

Equation (2.3) serves to calculate the W/B of the icosahedron for any altitude below 20 km (65,000 ft) [17] and vacuum (partial or total). For a partial vacuum, the remaining internal pressure counteracts the external pressure, such that the pressure "felt" externally by the skin is given by:

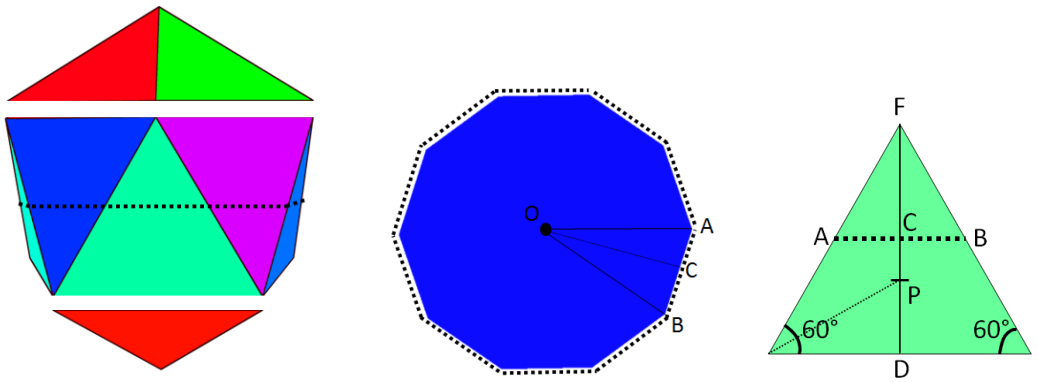
$$P_{applied} = P_{air,o} - P_{air,i} \tag{2.4}$$

where: $P_{applied}$ = air pressure applied to or 'felt' by the skin

If a total vacuum is achieved, the second term of Equation (2.3b) goes away since no air remains inside, such that $P_{applied}$ equals the Sea Level (SL) pressure.

Considering the icosahedron volume, one should note that a circumscribed sphere touches each of the vertices, such that the radius, r, is measured from the center to any vertex. In addition, an inscribed sphere with radius, r_i , touches each triangle at the centroid. Now consider a mid-plane perpendicular to an imaginary line drawn in between opposite vertices, extracted from Figure 5a and shown in Figure 5b by the dotted line, where A and B are two vertices on the mid-plane, as shown in Figure 5b. Then, $r = OA = OB$, where O is the icosahedron center. The center cutout shown in Figure 5a has 10 faces around, therefore the angle OAB is 36°. Then,

$$OC = AC\ cot18° = BC\ cot18° \tag{2.5}$$



(a) Partitions (b) Mid-plane (c) Equilateral Triangle

Figure 5: Icosahedron Decomposition

Also consider one of the equilateral triangular faces with the points A, B, C, D, F and P, as shown in Figure 5c. Then, $r_i = OP$. Given the triangle's edge length, the height (DF) can be obtained with the Pythagorean theorem, as follows:

$$l_{beam}^2 = DF^2 + \left(\frac{l_{beam}}{2}\right)^2 \Rightarrow DF = h = \frac{\sqrt{3}}{2} l_{beam} \tag{2.6}$$

where:

h = height of the triangle

l_{beam} = edge length

Now, knowing that the the angle between sides is 60°, the centroid (DP) is given by:

$$DP = \frac{l_{beam}}{2} \tan 30^\circ = \frac{\sqrt{3}}{6} l_{beam} \tag{2.7}$$

Since it is an equilateral triangle, $AC = l_{beam}/4$, Equation (2.5) becomes:

$$OC = \frac{l_{beam}}{4}\, cot 18^\circ \tag{2.8}$$

Then, given that OCP is a right triangle, the Pythagorean theorem can be used in Equation (2.6), Equation (2.7) and Equation (2.8),

$$\begin{aligned} OP^2 &= OC^2 - CP^2 = OC^2 - (CD - DP)^2 = OC^2 - (h/2 - DP)^2 \\ &= \tfrac{l_{beam}^2}{16} \cot^2 18^\circ - \left(\tfrac{\sqrt{3}}{4} l_{beam} - \tfrac{\sqrt{3}}{6} l_{beam}\right)^2 = l_{beam}^2 \left(\tfrac{\cot^2 18^\circ}{16} - \tfrac{1}{48}\right) \end{aligned}$$

$$r_i = OP = l_{beam} \sqrt{\frac{\cot^2 18^\circ}{16} - \frac{1}{48}} \approx 0.7558 l_{beam} \tag{2.9}$$

Equation (2.9) provides the radius of the inscribed sphere with respect to the edge length. In the same fashion, the radius of the circumscribed sphere (icosahedron radius) is obtained with respect to the edge length:

$$OF^2 = OP^2 + FP^2 = OP^2 + (h - DP)^2$$

$$OF^2 \approx (0.7558 l_{beam})^2 + \left(\tfrac{\sqrt{3}}{2} l_{beam} - \tfrac{\sqrt{3}}{6} l_{beam}\right)$$

$$r = OF \approx 0.9511\ l_{beam} \tag{2.10}$$

Finally, the volume of the icosahedron can be represented by 20 pyramids of height r_i, such that:

$$A = \frac{\sqrt{3}}{4} l_{beam}^2 \tag{2.11}$$

$$V_i = 20\left[\left(\frac{1}{3}A\right) r_i\right] = \frac{5}{12}\left(3 + \sqrt{5}\right) l_{beam}^3 \tag{2.12}$$

where:

A = area of an equilateral triangle

V_i = icosahedron internal volume

Equation (2.10) and Equation (2.12) were verified with Reference 29 and Reference 51. Both equations are used to develop the W/B equations for the frame and the skin in terms of the icosahedron radius.

It is desirable to select geometric properties that allow for the icosahedron to achieve buoyancy, i.e., $W/B < 1$. But V_r depends on the analysis results and the geometric properties are needed in order to establish the model. Nonetheless, assuming that:

- No internal volume is lost due to the skin displacement or internal components,
- The material densities remain constant throughout the analysis,
- A total vacuum is achieved, and
- The structure is vacated at SL altitude,

provides a method of estimating the geometric properties of each component given a desired W/B.

2.2.1 Skin W/B.

Provided that the skin is composed of 20 equilateral triangles, the volume can be represented using Equation (2.11):

$$V_{skin} = 20\left(\frac{\sqrt{3}}{4} l_{beam}^2\right) t_{skin} = 5\sqrt{3} t_{skin} l_{beam}^2 \tag{2.13}$$

where: t_{skin} = icosahedral skin thickness

Using the assumptions previously stated[7] and considering the skin separate from the frame, the skin thickness can be derived for a skin W/B set value. Equation (2.1) then reduces to:

$$\frac{W}{B}_{skin} = \frac{V_{skin}\,\rho_{skin}}{V_{internal}\,\rho_{air}} \tag{2.14}$$

where: ρ_{air} = density at SL

Substituting Equation (2.10) in Equation (2.12) and Equation (2.13), and then combining them with Equation (2.14), the skin thickness is:

$$t_{skin} \approx \frac{r\,\rho_{air}\,(W/B_{skin})}{3.77523\,\rho_{skin}} \tag{2.15}$$

Equation (2.15) is used in all the analyses related to the skin to estimate a t_{skin} that provides a desired W/B.

2.2.2 Frame W/B.

As done for the skin, the geometric properties of the frame can be obtained for a specific W/B. In this case, a circular cross-section is selected for frame beams; therefore, two geometric parameters arise: the beam radius and beam thickness. the beams radius

[7] Assuming that there is no internal volume loss due to the skin displacement or internal components, that the the skin displacement causes no change in its volume, that a total vacuum is achieved and that the structure is vacated at SL altitude.

and thickness. A pictographic representation is shown in Figure 6. In order to solve for them, the following relationship is established:

$$t_{beam} = c\, r_{beam} \quad for \quad 0 < c \leq 1 \tag{2.16}$$

where:

r_{beam} = beam radius

t_{beam} = beam thickness

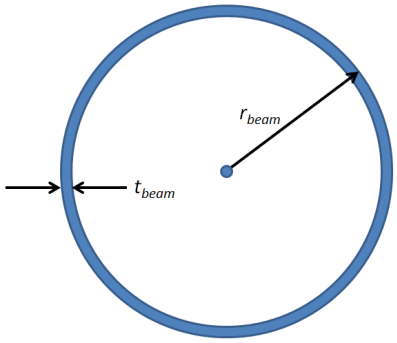


Figure 6: Beam Profile

With Equation (2.16) and knowing that the icosahedral frame is composed of 30 beams, the volume can be obtained in terms of c as follows:

$$V_{frame} = 30\pi\left(2r_{beam}t_{beam} - t_{beam}^2\right)l_{beam} = 30\pi r_{beam}^2\left(2c - c^2\right)l_{beam} \tag{2.17}$$

Once again, using the assumptions previously stated[8] and considering the skin separate from the frame, Equation (2.1) reduces to:

[8]Assuming that there is no internal volume loss due to the skin displacement or internal components, that the the skin displacement causes no change in its volume, that a total vacuum is achieved, and that the structure is vacated at SL altitude.

$$\frac{W}{B}_{frame} = \frac{V_{frame}\,\rho_{frame}}{V_{internal}\,\rho_{air}} \tag{2.18}$$

Substituting Equation (2.10) in Equation (2.12) and Equation (2.17), and then combining them with Equation (2.18), the beam radius for $0 < c \leq 1$ is:

$$r_{beam} \approx r\sqrt{\frac{(W/B_{frame})\,\rho_{air}}{39.0742\,(2c - c^2)\,\rho_{frame}}} \tag{2.19}$$

Equation (2.19) is used in all the analyses related to the frame to estimate a r_{beam} that provides a desired W/B. Note that for a solid beam, $c = 1$.

2.2.3 Icosahedron W/B.

It is important to include all the effects -altitude, vacuum (partial or total), and volume reduction- in the W/B calculation. Substituting Equation (2.13) and Equation (2.17) in Equation (2.3b), which accounts for all these effects, results in:

$$\frac{W}{B} = \frac{5\sqrt{3}t_{skin}l_{beam}^2\,\rho_{skin} + 30\pi r_{beam}^2\left(2c - c^2\right)l_{beam}\,\rho_{frame}}{(V_i - V_r)\left(\frac{P_{air,o}}{RT_{air,o}}\right)} + \frac{P_{air,i}}{P_{air,o}}\frac{T_{air,o}}{T_{air,i}} \tag{2.20}$$

Then substituting Equation (2.10) and Equation (2.12) in Equation (2.20) and simplifying, results in:

$$\frac{W}{B} = \frac{9.5745t_{skin}\,r^2\,\rho_{skin} + 99.098\left(2c - c^2\right)r_{beam}^2\,r\,\rho_{frame}}{[2.5362r^3 - V_r]\left(\frac{P_{air,o}}{RT_{air,o}}\right)} + \frac{P_{air,i}}{P_{air,o}}\frac{T_{air,o}}{T_{air,i}} \tag{2.21}$$

Equation (2.21) along with Equation (2.4) are used to calculate the resulting W/B as the structure deforms due to the applied pressure. Note that Equation (2.21) is applicable for any icosahedron radius and any skin and frame materials, as long as the following criteria is meet:

- Material homogeneity: both frame and skin densities remain constant within each part.
- Geometric homogeneity: both frame and skin geometry, r_{beam} and t_{skin}, remain constant through the icosahedron.
- Moderate dimensions: if r_{beam} and t_{skin} are large compared to each characteristic length, the icosahedron radius, r, needs to be adjusted individually to compensate. At this point, the W/B needs to be reconsidered along with the FEA technique.
- The altitude considered is less that 65,000 ft.

2.3 Air Properties with Altitude

Creating an internal vacuum in any enclosed structure that is exposed to the environment, generates pressure forces on the structure's external surfaces to try and balance the pressure difference. The amount of external pressure depends on the amount of vacuum generated and the altitude. The barometric formula shown below indicates the changes in air pressure versus altitude.

$$P = P_o \exp\left(-\frac{g}{RsT_o}H\right) \tag{2.22}$$

where:

H = altitude

P_o = SL pressure

Rs = specific gas constant

T_o = temperature at SL

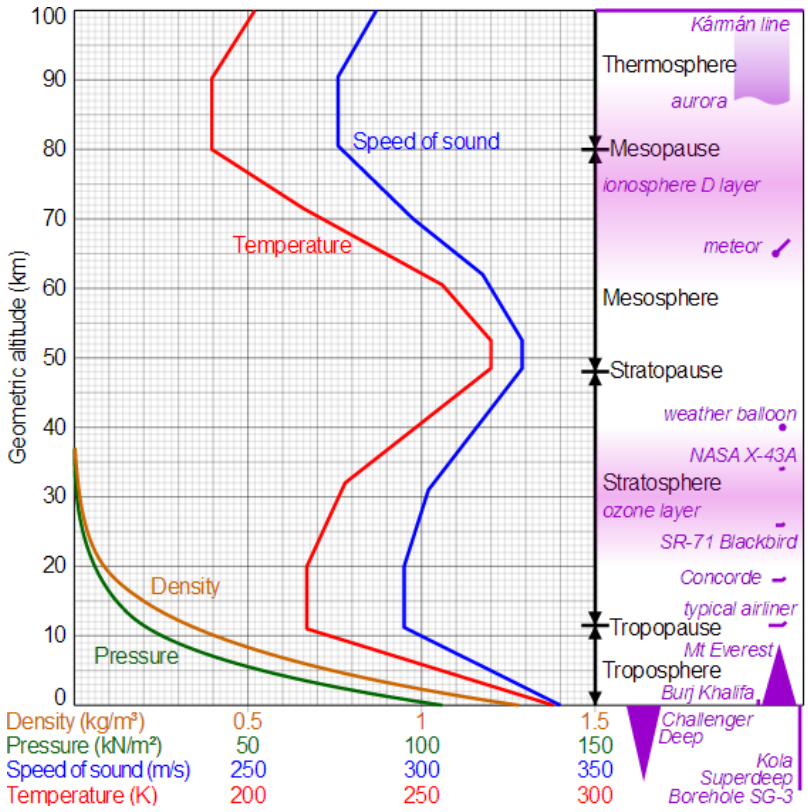


Figure 7: Air Properties vs Altitude [50]

Note that pressure changes exponentially with altitude. The changes in pressure and altitude are graphed in Figure 7, along with changes in density and temperature. Note that maximum pressure and density are achieved at SL. As seen in Equation (2.1), having the lowest denominator is ideal since it improves buoyancy, and the latter is maximized when all the air volume is extracted, considered a perfect vacuum, and the structure is at SL. Therefore, the standard air properties, pressure (101,325 Pa), density (1.2041 kg/m^3) and temperature (288.15 K), are used in this thesis for the calculations of W/B and applied load [1].

2.4 Nonlinear Analysis

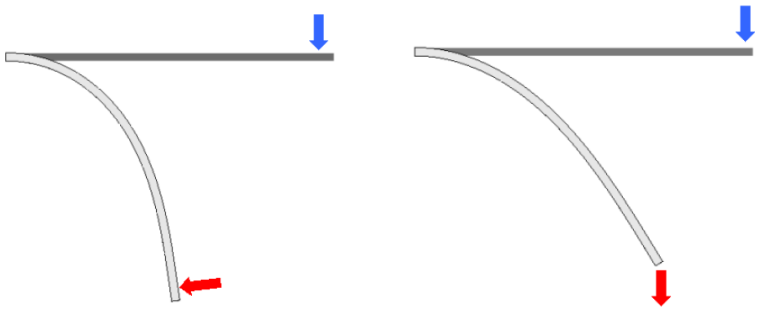
Historically, the use of linear analysis tools to describe the behavior of a body under load has been preferred. These tools provide an acceptable approximation for most real-life problems. Their availability and simplicity make them an attractive choice over the complexities brought by nonlinear analysis tools. But there are problems that require nonlinear analysis in order to capture the 'true' structural behavior. In order to solve such problems, algorithms have been developed and included in computer software, most commonly referred as FEM, which use the governing equations along with numerical solution methods to solve for the structural response. One of the most common solution techniques is the Newton Raphson. Summaries of nonlinear relationships in structure behavior and the Newton Raphson method follow.

2.4.1 Nonlinearities.

The term "stiffness" is a property that characterizes the structural response of a body subjected to loading. In general, the structure's stiffness changes as it is being deformed. But if small deformation occurs, the structure can retain the stiffness that it had prior to loading; this is what is characterized as linear behavior. When large deformation occurs, the structural stiffness changes, causing nonlinearities [4]. The nonlinearities are described by the governing equations. Geometric nonlinearity, found in the strain displacement relationships and the equilibrium equations, is characterized by a change in geometric shape. Material nonlinearity, found in the constitutive laws, usually result from structure straining past the yielding point [42, 21]. Therefore, these two nonlinearities are treated independently. In this thesis, geometric nonlinearity is particularly important since large deflections are expected. Large deflections come as a byproduct of the weight

restrictions given by the nature of LTAV, and drive the geometric parameters of the icosahedron. Therefore, only geometric nonlinearity is considered.

When using FEA to evaluate a structure that exhibits nonlinear behavior, several considerations need to be taken into account. First, the force direction. Large deformation can cause a change in force direction. Two cantilever beams subjected to a concentrated force are shown in Figure 8. In Figure 8a, the force remains normal to the beam as it displaces; this force is called a follower force. In Figure 8b, the force retains its original direction as the beam displaces; this force is called a non-follower force. In the same way that large deformations affect the force direction, the force direction affects a structure with large deformations. FEA software usually provides the option of selecting the type of force you wish to use for the model.



(a) Follower Force (b) Non-follower Force

Figure 8: Follower and Non-Follower Forces. A follower force changes its direction during the process of deformation and remains normal to the deformed beam (left). A non-follower force retains its original direction (right) [4].

A good example is the icosahedron itself. The icosahedron is subjected externally to the atmospheric pressure, and pressure always acts normal to the surface (follower force). While linear analysis considers no geometric changes, realistic analysis includes geometric nonlinearity to account for those changes.

Another consideration is stiffness change. An example is a flat shell (capable of carrying both bending (out-of-plane) and membrane (in-plane) stiffnesses) subjected to a pressure load. Initially, the flat shell resists the pressure only with its bending stiffness. After some deformation has developed, the shell acquires membrane stiffness, stiffening the shell as the pressure increases. In this case, nonlinear analysis is required regardless of having small or large deformations. Note that the icosahedral skin, composed of initially flat triangles, depends on nonlinearity to develop membrane stiffness.

Other geometrically nonlinear considerations include buckling (discussed in Section 2.6) and post buckling behavior, supports that cause changes in the structure's stiffness, and contact problems.

2.4.2 The Newton Raphson Method.

The Newton Raphson method is an iterative technique that solves nonlinear equations. In the case of FEA, this technique solves the nonlinear static equilibrium equations that govern the structural behavior of a model by diving the loading in small steps and finding the solution path in an incremental fashion. The following discussion, based on the work of A.N. Palazotto and S.T. Dennis [42, 70, 131-134], summarizes such a method for a one-dimensional case.

The nonlinear static equilibrium equations resulting from a finite element discretization are in the form of:

$$K_T \Delta q = -F(q)$$
$$\left[K + N_1(q) + N_2\left(q^2\right)\right]\Delta q = -\left[K + \tfrac{1}{2}N_1(q) + \tfrac{1}{3}N_2\left(q^2\right)\right]q + R \tag{2.23}$$

where:

K = linear stiffness matrix

K_T = tangent stiffness matrix

N_1 = linear function of q

N_2 = quadratic function of q

q = nodal displacements vector

R = nodal loading vector

Given that at the first load increment, R_1, $q = 0$, Equation (2.23) reduces to a linear equation:

$$Kq_1 = R_1 \tag{2.24}$$

such that $K_T = K$, the slope of the load versus displacement curve at $q = 0$; this would be the first iteration. In the second iteration, q_1 is substituted in Equation (2.23) to solve for the increment Δq_1, as shown below:

$$\left[K + N_1(q_1) + N_2\left(q_1^2\right)\right]\Delta q_1 = -\left[K + \frac{1}{2}N_1(q_1) + \frac{1}{3}N_2\left(q_1^2\right)\right]q_1 + R_1 \tag{2.25}$$

The right side of Equation (2.25) is the residual force vector ΔR_1 that is left by the second iteration. The displacement q is then updated as follows,

$$q_2 = q_1 + \Delta q_1 \tag{2.26}$$

Iteration continues until the residual force vector ΔR_n becomes small, signifying that the equilibrium equations are satisfied for the first load increment. The process is repeated

for the next load increment, R_2, as shown in Figure 9 until convergence. Points *A* and *B* represent the solutions for load increments 1 and 2.

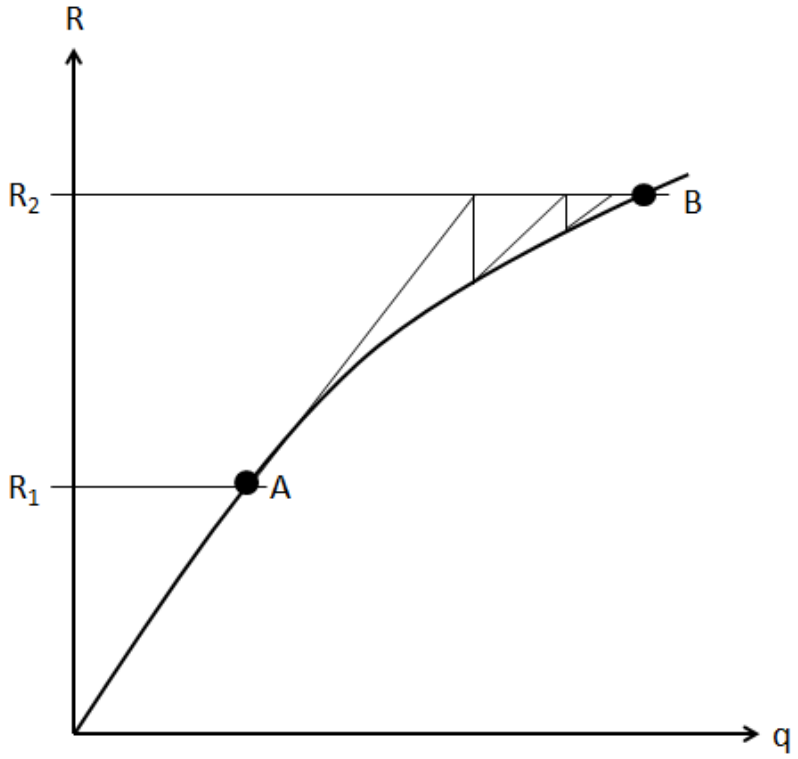


Figure 9: Newton Raphson Algorithm: second increment [42, 133]

In reality, discretized models have multiple DOF, therefore instead of having one load versus displacement curve, there are as many curves as DOF. For such cases, a global convergence criterion is established, such as:

$$\frac{\sqrt{\sum_i (q_r^i)^2} - \sqrt{\sum_i \left(q_{r-1}^i\right)^2}}{\sqrt{\sum_i \left(q_1^i\right)^2}} x100\% \leq TOL \qquad (2.27)$$

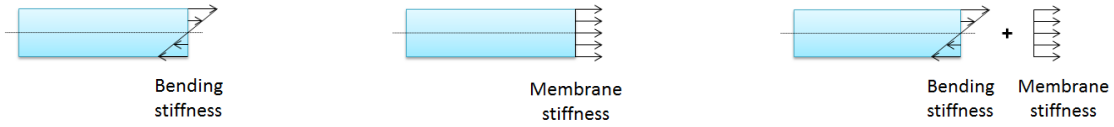
where q_r^i, q_{r-1}^i and q_1^i are the elements of q ($i = 1$ to n DOF) for the rth, $(r-1)th$ and first iterations for a given load increment, and TOL is user defined, typically taken as 0.1%. A drawback of the Newton Raphson method is its inability of crossing points where the load versus displacement slope is zero. For these cases, a displacement driven method, or the Riks method [12, Ch. 6.2.4], which varies the load and displacement simultaneously, is used.

2.5 Membranes, Plates and Shells

There are many applications using plates, membranes and shells components in structures. Therefore published solutions are largely available for common geometries such as the circle, rectangle, sphere and cylinder.

Literature commonly refers to plates and shells as structural parts that are initially flat and curved, respectively, where their thickness is much smaller than their characteristic length [49, 1]. Shell are characterized by both bending, out-of-plane, and membrane, in-plane, stiffnesses; therefore used for, among many others, cylindrical and spherical applications. On the other hand, plates are primarily characterized by their bending stiffness, therefore used for applications where a load is applied normal to a surface in which small deformations occurs. The membranes then result from two scenarios: plates with in-plane applied forces and thin plates with out-of-plane applied forces. For the first scenario, in-plane forces produce no bending stiffness, developing all its resistance through its membrane stiffness. For the second scenario, the plate becomes thin enough that the bending stiffness becomes negligible. At that point, large deformations tend to occur and membrane stiffness is developed as it deforms. All three types of parts consider displacement values from the middle surface. A pictographic

representation of the stiffness distributions for the plate, membrane and shell are shown in Figure 10. Note that while the shell has both bending and membrane stiffnesses (Figure 10c), the plate and membrane have bending (Figure 10a) and membrane (Figure 10b) stiffnesses only, respectively.



(a) Plate (b) Membrane (c) Shell

Figure 10: Stiffness Distribution of Plates, Membranes and Shells

Considering one of the equilateral triangles of the icosahedron and the importance of weight for an LTA application, researchers were initially focused in solutions for flat membranes subjected to a uniform distributed load (also called uniform pressure). Since no membrane solution for an equilateral triangle was found[9], circular and square solutions were reviewed for the purpose of FEM validation.

S. Timoshenko and S. Woinowsky-Krieger suggest membrane solutions for both circular and square geometries with large deflections. The circular solutions are derived from the equilibrium equations developed for a circular plate that also carries membrane forces, but neglecting the terms that relate to bending [45, 402], assuming the vertical deflection takes the form:

$$w = w_o\left(1 + \frac{r_c^2}{a_c^2}\right)^2 \tag{2.28}$$

[9] A.C. Ugural has a plate solution for equilateral triangle subjected to a uniform pressure [49, 98-100].

where:

a_c = circle's radius

r_c = radial position at which the displacement is evaluated ($0 \leq r \leq a_c$)

w = vertical displacement

w_o = center displacement

and then solving for w_o iteratively. The circular displacement and stress solutions are then given by Equation (2.29) and Equation (2.30) [45, 400-404]:

$$w = w_o\left(1 + \frac{r_c^2}{a_c^2}\right)^2 \quad where \quad w_o = 0.662 a_c\left(\frac{P a_c}{Et}\right)^{1/3} \tag{2.29}$$

$$S_o = 0.423\left(\frac{EP^2 a_c^2}{t^2}\right)^{1/3} \tag{2.30}$$

where:

E = modulus of elasticity

P = applied pressure to the surface

S_o = stress at the center

t = membrane thickness

These solutions are based on a Poison's ratio, ν, equal to 0.25 and fixed BC along the edges, and assume linear elastic material properties. Both equations show that the deflections and stresses vary as the cube root of the pressure. In the case of the square solution, Timoshenko and Woinowsky-Krieger use energy methods along with assumed displacement fields to find the solution of a square membrane. They define the strain energy solely due to stretching of its middle surface as [45, 419-420]:

$$V_s = \frac{1}{2}\int\int \left(N_x\varepsilon_x + N_y\varepsilon_y + N_{xy}\gamma_{xy}\right)dxdy \tag{2.31}$$

where:

V_s = strain energy

N_x = membrane force in the x direction

N_y = membrane force in the y direction

N_{xy} = membrane force in the xy direction

ε_x = strain in the x direction

ε_y = strain in the y direction

ε_{xy} = strain in the xy direction

Equation (2.31) is then put in terms of displacement fields, followed by inserting the assumed displacement fields shown in Equation (2.32), and using the principle of virtual displacements to solve for w_o and c_s.

$$u = c_s \sin\frac{\pi x}{2a_s}\cos\frac{\pi y}{2a_s}; \quad v = c_s \sin\frac{\pi y}{2a_s}\cos\frac{\pi x}{2a_s}; \quad w = w_o \cos\frac{\pi x}{2a_s}\cos\frac{\pi y}{2a_s} \tag{2.32}$$

$$c_s = 0.147\frac{w_o^2}{a_s} \quad \& \quad w_o = 0.802a_s\left(\frac{Pa_s}{Et}\right)^{1/3} \tag{2.33}$$

$$S_o = 0.396\left(\frac{Eq^2a_s^2}{t^2}\right)^{1/3} \tag{2.34}$$

where:

$u,\ v,\ w$ = x, y, out-of-plane displacements

a_s = half the edge length

These solutions are also based on a $\nu = 0.25$ and fixed BC along the edges. Geometric definitions are shown in Figure 11. Note that the maximum displacement occurs at the center, when $x = y = 0$. As with the circular solution, both equations show that the deflections and stresses vary as the cube root of the pressure. Furthermore, comparison between the circle and square's equations shows that the center displacement and stress vary only by 5% and 2%, respectively, when the circle diameter equals the square edge length.

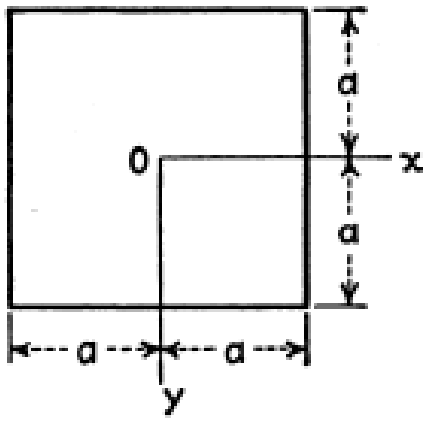


Figure 11: Square Membrane Geometrical Definitions [45, 420]

P. Seide found an alternate solution for the square membrane by iteratively solving the Fȯppl's large deflection equations and Airy's stress function [47]. He found that the center displacement is given by:

$$w_o = 0.2866t\left[\frac{q}{E}\left(\frac{b}{t}\right)^4\right]^{1/3} = 0.722a\left(\frac{qa}{Et}\right)^{1/3} \tag{2.35}$$

where: b = membrane's edge length ($b = 2a_s$, as previously defined in Timoshenko's solution). Seide's solution assumes a $\nu = 0.3$. Disregarding the 0.5 difference in Poison's ratio between solutions[10], Seide's solution (Equation (2.35)) predicts a center displacement that is 90% of the one predicted by Timoshenko and Woinowsky-Krieger's solution (Equation (2.32)). All these solutions are used to validate the FEM and run convergence studies. See Section 3.4 for more details.

[10]Changes of ν between 0.1 and 0.4 only affects the membrane solution by 10%. See Section 3.5.3 for more details.

2.6 Material, Buckling and Collapse Failures

Different failure modes exist on a structure subjected to loading, including material, buckling and collapse. Material failure is related to the material's capacity to withstand stress, in other words, its strength. Buckling failure, on the other hand, is related to the structure's geometry. Collapse also relates to the structural geometry, but is characterized by global failure. All failure modes are related to the structure's stiffness, which is a combination of the stiffnesses provided by the material and geometrical shape.

2.6.1 Material Failure.

Material failure comes as a byproduct of the stresses produced by the applied load on the structure. The point at which the material fails can be defined as when the stresses reach the yielding point, the ultimate point or somewhere in between. Ductile materials exhibit both points, where plasticity occurs in between the two points. On the other hand, brittle materials tend to either lack the yielding point or the same is very close to the ultimate point.

Considering ductile materials, there are several failure criteria that predict the failure of a structure, including the Von Mises yield criterion. The von Mises criterion, also known as the maximum distortion energy criterion, states that failure occurs when the energy of distortion reaches the same energy for yielding in uniaxial tension. The following equations shows the von Mises criteria:

$$\sqrt{\frac{1}{2}\left[(\sigma_{11}-\sigma_{22})^2+(\sigma_{22}-\sigma_{33})^2+(\sigma_{11}-\sigma_{33})^2+6\left(\sigma_{23}^2+\sigma_{31}^2+\sigma_{12}^2\right)\right]} \equiv S_f/SF \quad (2.36)$$

where:

SF = safety factor

S_f = design failure point

σ_n = stress in the directions $n = 1, 2, 3$

Material failure, either by yielding or breaking, becomes an important consideration when evaluating both the icosahedral skin and frame, particularly in the skin/frame connections where stress concentrates.

2.6.2 Buckling.

Buckling can be defined as an instability phenomena where a structure is unable to recover from its initial state of equilibrium after been disturbed. In general, a loaded structure is said to be in state of equilibrium if for all displacements from the equilibrium state, restoring forces arise such that the structure moves back to equilibrium. Consider the simple case of fixed end column, as shown in Figure 12. A force P applied at the top but away from the centroid causes a moment about O which tends to bend the column; on the other hand, elastic forces created by its stiffness tend to restore it to its equilibrium position, remaining statically stable. As P increases, there is a point at which the bending moment is so high that the column's stiffness is insufficient to restore it, becoming unstable. At that point the column has buckled and P becomes the critical load. Linear theory, derived

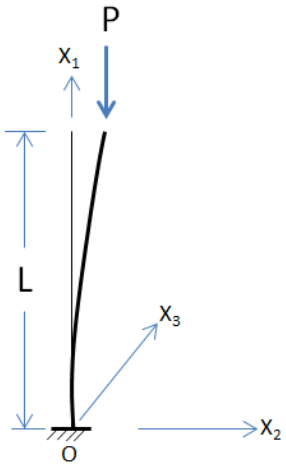


Figure 12: Fixed Column Subjected to a Concentrated Load

with the Euler-Bernoulli beam equation, shows that the critical load is given by Equation (2.37) [46, 496]:

$$P_{cr} = \frac{\pi^2 EI}{4L^2} \tag{2.37}$$

where:

I = area moment of inertia

L = column length

P_{cr} = critical load

Once buckling occurs on a structure, its stiffness changes thus causing nonlinear response, called post-buckling. The post-buckling response ends once the structure is not capable of carrying any load. At this point, the structure is said to have collapsed.

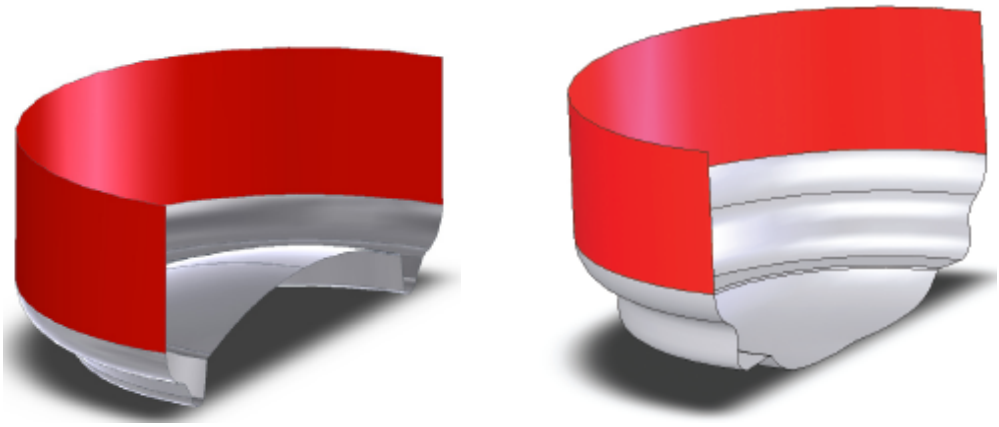
2.6.3 Collapse.

Collapse is a geometric phenomenon where the structure suddenly loses its capacity to resist the applied loading and its geometry distorts; at that point the structure becomes globally unstable. Collapse can result from 'local' buckling, e.g, buckling of some icosahedral frame[11] beams triggered by unsymmetrical loading causes the whole icosahedron to loose its stiffness. Numerically, structural collapse can be characterized as the moment at which the structure shows a negative stiffness and it must release strain energy in order to remain in equilibrium [12, Ch. 6.2.4].

The behavior of a structure close to its collapse point usually displays nonlinear nature. The Newton Raphson method described in Section 2.4.2 works well for nonlinear problems, but it is unable to cross the buckling points. The Riks method is recognized for

[11]The term 'frame' most commonly refers to a structure composed of an array of beam members (members that resist both axial and bending loads), which is the case of the icosahedral frame.

its capacity to trace the buckling and post-buckling behavior up to the structural collapse point. The importance of tracing the post buckling behavior can be easily seen by considering the snap-through of a aluminum can. The snap-through behavior of an aluminum can is shown in Figure 13, both prior to buckling (Figure 13a) and after buckling (Figure 13b). Note that the can still retains its load-bearing capacity after buckling.



(a) Pre-buckling (b) Post-buckling

Figure 13: Aluminum Can Snap Through [4]

A common aeronautical application is the use of thin walled stringer stiffened panels in fuselage structures. R. Degenhardt, H. Klein, A. Kling, H. Temmen and R. Zimmermann studied the behavior of a stringer stiffened carbon-fiber-reinforced polymer panel subjected to quasi-static compressive loading, the type of loading wing panels are subjected to, using both experimental and FE methods [15]. In the FEA, they superimpose the mode shapes from a linear buckling analysis into the initial geometry to

create artificial ‘imperfections’, and then use the nonlinear Newton Raphson technique with adaptive stabilization. In contrast, they use optical digitizing to measure the imperfections of the real panel when conducting experiments. Scaled load versus shortening curves for the panel are shown in Figure 14. The panel shows buckling at about a scaled shortening of 2, and collapse is seen at about a scaled shortening of 3.5. Since the Newton Raphson technique is being used, collapse is not captured (see the ABAQUS/Standard curve). Collapse of such panels usually results from buckling of their stiffeners. The buckling of the skin depends on its thickness and radius of curvature. As the radius of curvature increases, more panel bending stiffness shifts to membrane stiffness, increasing its critical pressure, since buckling results from the bending effect.

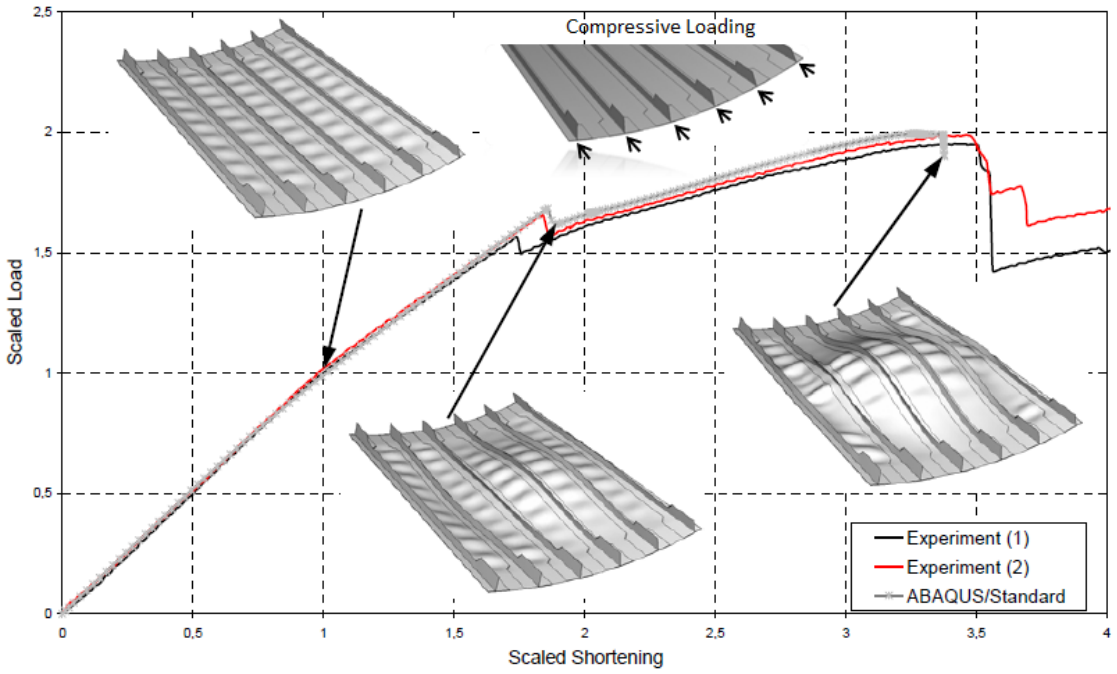


Figure 14: Panel Collapse: Load-shortening Curves [15]

When analyzing the icosahedron, locating the collapse point is particularly important. As pressure is transferred from the skin to the frame, each frame member is subjected to both axial and transverse loads, becoming vulnerable to buckling. On the other hand, the icosahedral skin does not exhibit collapse in the form of buckling for two reasons. First, the frame/skin icosahedral configuration subjected to a uniform pressure only produces tensile stress on each one of the skin triangles. Second, having the skin act as a membrane eliminates the bending stiffness, ergo, no bending moment would be present to develop the compressive foces that cause buckling.

2.7 Materials Research

Designs demand different material characteristics that depend on their applications and requirements. In case of an LTAV, the most important characteristic tends to be density, but stiffness and strength are also relevant factors. On the other hand, the pressure difference created by an internal vacuum on a vacuum LTAV puts significant strain on the structure, therefore maximizing stiffness and strength while minimizing density is desired.

Establishing the effects of different materials on the performance of a structural component allows for an optimal selection of materials. Given a design objective, a performance index is developed that relates the structural response of a component to the material characteristics. These indexes are function specific, therefore they try to maximize an aspect of the component's performance. Consider the icosahedral skin/frame subjected to the pressure created by the internal vacuum. Part of that pressure 'felt' by the skin is transferred to the frame as a distributed load in each beam. If each beam is treated as a separate component, material indexes can be developed to guide the

material selection of the icosahedron. Now consider a simple supported beam subjected to a distributed load, as shown in Figure 15. The objectives in this case are to minimize the beam's weight while maximizing its stiffness and strength. Let consider the first objective: stiffness. One can relate the displacement and weight of the beam to its stiffness as follows:

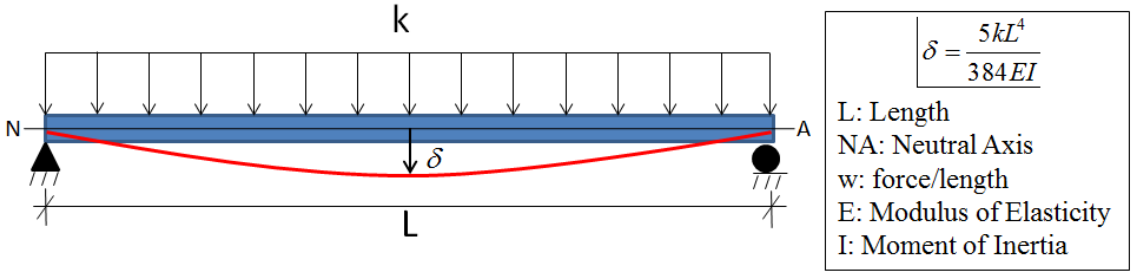


Figure 15: Simply Supported Beam with Distributed Load [23]

$$S = \frac{k}{\delta} = \frac{384EI}{5L^4} \tag{2.38}$$

$$W = AL\rho \tag{2.39}$$

where:

A = profile area

I = moment of inertia

L = beam length

S = stiffness

k = distributed load

W = beam weight

ρ = density of the material

δ = maximum displacement

Considering a circular profile, $I = \frac{\pi}{4} r_{beam}$ and $A = \pi r_{beam}^2$. Then substituting I in Equation (2.38) and solving for r_{beam}:

$$r_{beam} = \left(\frac{20L^4 S}{384\pi E} \right)^{1/4} \tag{2.40}$$

Then, substituting A in Equation (2.39) where r_{beam} is given byEquation (2.40), results in:

$$W = \pi L \rho \sqrt{\frac{20L^4 S}{384\pi E}} = e \frac{\rho}{E^{1/2}} \tag{2.41}$$

where: e = constant

Equation (2.41) establishes an index that relates the beam's weight and its stiffness. In order to minimize the weight, the ratio $(E^{1/2}/\rho)$ needs to be maximized. This index is called the specific stiffness, and it applies to other loading types. For example, the maximum displacement solution for a beam subjected to a concentrated load, P, is $(PL^3)/(48EI)$. Since this loading also produces bending, the index remains the same. Therefore, the specific stiffness index is independent of the load type, as long as that load produces bending. Furthermore, the BC selection, e.g., fixed, simply supported or cantilever, does not affect the index either. Now considering strength, the stress due to bending is given by:

$$\sigma_b = \frac{M_b y}{I} \tag{2.42}$$

where:

M_b = bending moment

y = distance from the neutral axis

σ_b = bending stress

Considering that the maximum stress occurs at $y = r_{beam}$, and substituting I in Equation (2.42), produces:

$$\sigma_f = \frac{M_b y}{I} = \frac{4M_b}{r_{beam}^3} \tag{2.43}$$

where: σ_f = design failure stress

Substituting A and Equation (2.43) in Equation (2.39), results in:

$$W = \pi L \rho \left(\frac{4M}{\sigma_f}\right)^{2/3} = e\frac{\rho}{\sigma_f^{2/3}} \tag{2.44}$$

Once more, in order to minimize the weight, the ratio ($\sigma_f^{2/3}/\rho$) needs to be maximized. This ratio is called the specific strength index. Is important to state that the index is dependent on the beam profile. For example, a beam with a rectangular profile with fixed height and free width, has a specific stiffness index given by E/ρ [6].

Considering the icosahedral skin, a high specific stiffness is critical to ensure that the loss in internal volume resulting from the skin deflection does not result in considerable loss of buoyancy. A high specific strength in both the frame and the skin prevents failure in connection areas where the stress concentrates.

Other designs factors include manufacturability and diffusivity. The material properties that yield the desired specific stiffness and strength are, more often than not, tied to the manufacturing process whereas the diffusivity tends to be a material property. Both factors are usually overcame by new manufacturing technologies and coatings that prevent diffusion. Therefore, this literature review focuses on finding materials that

produce high specific stiffness and strength, such that a relationship between current (those found on specific materials) and feasible (those needed for design feasibility) material properties can be established.

Focusing in specific stiffness and strength, various materials searches were made using MatWeb [7], an Internet-based material library. Materials that were selected for comparison are shown in Table 1 (empty spaces indicates values that were not available). Specific stiffnesses and strengths are included in columns 6 and 7-8, respectively, where S_y and S_u are the yield and ultimate stresses. For the purpose of this thesis, materials are assumed to have linear behavior. Therefore the material properties listed in Table 1 serve as reference points, and the models considered are related to material properties, not to the materials themselves. These indexes will be considered to establish various icosahedron models. See Section 3.7.2 for more details.

2.8 Vacuum Lighter than Air Vehicles Concepts

2.8.1 A. Akhmeteli and A.V. Gavrilin's Concept.

A. Akhmeteli and A.V. Gavrilin propose 'Layered Shell Vacuum Balloons' as an LTA design [2]. This patent (pending) starts off by detailing an analysis of an homogeneous spherical shell, as the one proposed by Lana (see Section 1.2). It starts off by providing mathematical proof that an homogenenous spherical shell buckles under atmospheric pressure for any known material, as no material has the needed specific stiffness (E/ρ^2)[12] of $4.5x10^5[m^5/(kg-s^2)]$ [2, 5] for a $\nu = 0.33$.

[12]The specific stiffness considered here is based on the buckling of a spherical shell, different from the one defined in Section 2.7, which is based on the bending of a beam.

Table 1: Material Selection

#	Name	$\rho\,(kg/m^3)$	ν	$E\,(GPa)$	$S_y\,(GPa)$	$S_u\,(GPa)$	$E(Pa)^{1/2}/\rho$	$S_y(Pa)^{2/3}/\rho$	$S_u(Pa)^{2/3}/\rho$
1	UHM Unidirectional Carbon Epoxy tubes	1870	0.3	440.00	-	3.73	355	-	1286
2	Zylon	1560	0.37	303.00	5.80	-	352	2069	-
3	Diamond like Carbon, or Diamond thin film [43]	2700	0.12	757.00	75.70	-	322	6627	-
4	Boron Monofilament	2570	0.33	400.00	-	3.60	246	-	914
5	Nanocyl NC7000 Thin Multi-Wall CNT [56]	1650	0.2	1000.00	10.00	30.00	606	2813	5851
6	Beryllium S-200, Tubing	1844	0.18	303.00	0.40	0.8	299	294	467
7	CoorsTek Boron Carbide Reaction-Bonded	2650	0.18	379.00	1.70	-	232	538	-
8	Duramold-2 Cast Aluminum Mold Plate	2800	0.33	738.00	0.14	-	307	96	-
9	BALTEK SB.150 Structural End-Grain Balsa	247	0.33	5.76	0.02	-	307	337	-
10	Honeywell Spectra 1000 Fiber	970	0.33	172.00	-	3.00	428	-	2144

ρ = density, ν = Poison's ratio, E = modulus of elasticity, S_y = yield strength, S_u = ultimate strength

It is important to realize that we have come far, material wise, from what we had when Lana proposed the use of copper to construct hollow spheres. The specific stiffness of Carbon Nanotubes (CNT) research grade is about $3.7x10^5[m^5/(kg-s^2)]$ [36], while the cooper specific stiffness is $0.018x10^5[m^5/(kg-s^2)]$ [35]; and even though CNT with such properties is not yet commercially available for shell type applications, such a difference in specific stiffness suggests that we are not far from a feasible point.

Akhmeteli and Gavrilin claim that a sphere constructed as a sandwich type structure where thin outer and inner layers are interconnected by a core layer provides enough specific stiffness to resist buckling due to an atmosphere of pressure while allowing for positive buoyancy using commercially available materials; set inner and outer layers would have approximately the same mass while the core layer would be significantly thicker.

Sandwich configurations indeed increase the specific stiffness of structures due to the relatively high stiffness of the external layers combined with a low stiffness, low density regardless, thick core that not only transmits shear but increases bending stiffness as more mass is located away from the neutral axis. A sandwich panel made of aluminum honeycomb core and skins impregnated with epoxy resin is shown in Figure 16.
Detailed analyses of the layered shell vacuum balloons concept are provided in Reference 2.

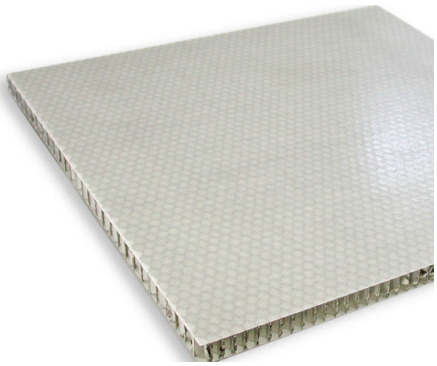
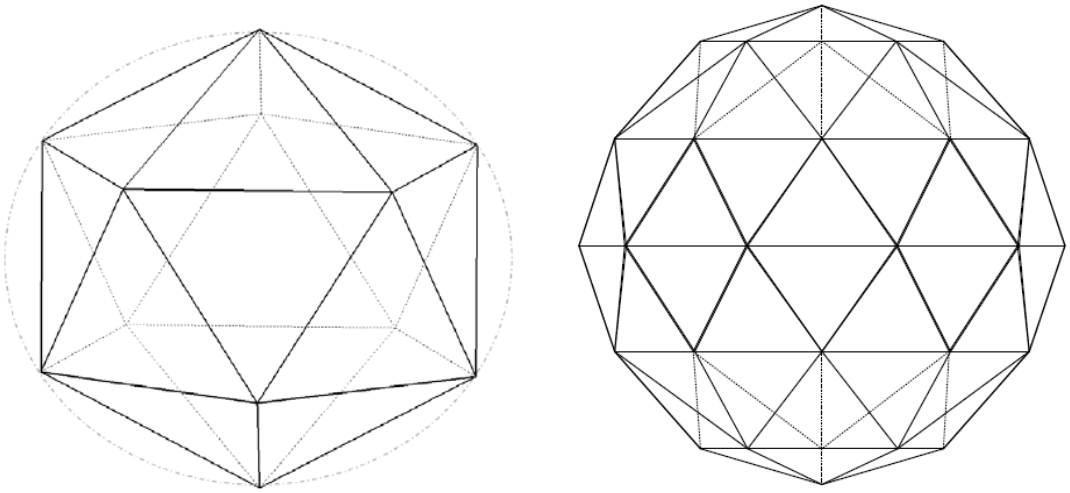


Figure 16: Sandwich Panel. Made of aluminum honeycomb core and skins impregnated with epoxy resin [39]

2.8.2 T.T. Metlen's Concepts.

T.T. Metlen presents various LTA concepts, including the icosahedron and the rotating cylinders. For the icosahedron, he performed an optimization of a 1.1 ft (0.33 m)



(a) Geodesic Sphere $f = 1$ (Icosahedron) (b) Geodesic Sphere $f = 2$

Figure 17: Geodesic Sphere [38, 47]

in radius geodesic sphere, in the form of a frame, where the objective was to minimize the weight to buoyancy of the frame, in other words, increase the buoyancy of the frame (detailed calculations of the weight to buoyancy ratio for the icosahedron are found in Section 2.2). The icosahedron is the simplest version of the geodesic sphere, as, shown in Figure 17a. If the edges of each triangle in the icosahedron are divided in two, creating three new vertices per triangle where all the vertices lie on the surface of a circumscribed sphere, each icosahedral triangle then becomes four triangles, as shown in Figure 17b. The geometric frequency is then defined as the number of divisions along the edges, such that the icosahedron represents a geodesic sphere of frequency 1, and the geodesic sphere becomes a 'perfect' sphere as the frequency tends to infinity. Metlen included the frequency as part of the optimization variables to evaluate its buoyancy effects. He showed that a frequency of 1 (the icosahedron) is the optimal configuration. Figure 18 shows the average and maximum stresses versus frequency. Note that for the

icosahedron, the maximum stress is equal to the mean stress. This result is attributed to the symmetry that is otherwise lost for frequencies greater than 1, which cause asymmetrical distribution of the pressure forces to the frame, increasing the maximum stress on the members [38, 111].

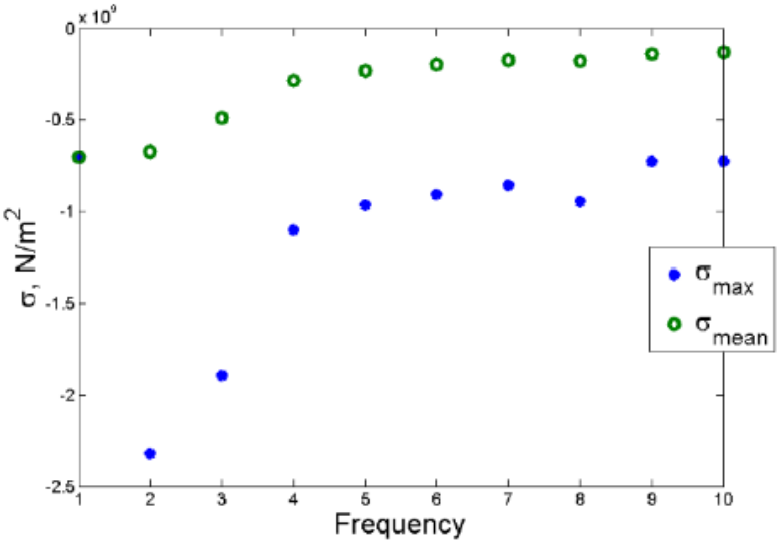


Figure 18: Stress versus Geometric Frequency of a Geodesic Frame [38, 112]

Metlen also considered the 'Rotating Cylinders'. The concept refers to having long thin skin cylinders rotate about their axis of symmetry, such that the centripetal force exerted by the skin would provide the additional stiffness needed to counteract the atmospheric pressure when an internal vacuum is created. He proposed the vehicle shown in Figure 19, composed of two smooth thin shell rotating cylinders mounted vertically into a gondola with propellers.

Considering the aerodynamics effects, the power requirements and its buoyancy, he found that for this vehicle to be feasible with a W/B less than 1, it would need to be 305-3100 meters long with cylinders radius of 1-10 meters for a W/B of 0.51, and it would need to be launched and operated above SL altitude where the drag is reduced by 99%, compared to the drag at SL.

Figure 19: Rotating Cylinders Vehicle [38, 56]

2.9 Summary

The homogeneous sphere has proven to be the best geometric shape for an LTA structure subjected to a vacuum, provided that a stiff, strong and light enough material exists, which as of today, it does not. That drives us to try alternate geometric shapes that provide the stiffness current materials lack. One of these geometries is the sandwich type, which Akhmeteli and Gavrilin claim to be feasible with today's materials. Another geometrical shape is the frame/skin icosahedral configuration, which Metlen researched as part of his thesis. See Section 2.8 for a summary of their proposed designs.

The lack of published literature on the icosahedron provides the opportunity to considered classical solutions and failure theory. Membrane solutions such as the one suggested by Timoshenko and Woinowsky-Krieger [45, 400-420], and Seide [47] provide a venue to compare against the FEM. Buckling and material failure theory provide background on the expected failure modes for the icosahedron, and its weakness points. Additionally, the material review establishes feasible ground for material properties that can be used to evaluate the icosahedron.

The structural knowledge can be combined with the LTA concept by the W/B. The W/B equations provide a venue of evaluating the buoyancy of the structure for any type of vacuum and altitudes up to 65,000 ft. Furthermore, reduced forms can be used to estimate the geometric properties needed in order to achieve the desired W/B. Given the geometric properties and theoretical background, nonlinear analysis can be performed in order to evaluate the structural behavior of the icosahedron, and used to calculate the W/B of the structure accounting for the skin deflection.

III. Model Development

3.1 Overview

The Oxford dictionary defines a model as "a three-dimensional representation of a person or thing or of a proposed structure ..." [41]. This definition brings an important question: what is a good representation? Answering this question requires running experiments or tests that verify the expected performance of the system being considered, and then making modifications as appropriate. But experimentation without proper modeling is usually infeasible and cost ineffective. Therefore, it is the modeler's job to try and provide the most accurate representation of the system. In order to do that, a validation process must be used. The difficulty of such a process depends on the complexity of the system and whether or not research on that system exist and is available.

In the case that the system does not exist, the validation process can be based on current systems that are related to the one considered. In case of the system considered in this thesis, an LTA icosahedral structure subjected to a vacuum, first: a vacuum LTAV is yet to be constructed, and second: limited research on the structural response of an icosahedron was found. On the other hand, the principles behind LTAV and the structural response of individual components is well understood. Therefore, the validation process was established based on the research found and analytical solutions of structural components that relate to the icosahedron.

The FEA relies in the discretization of a system to evaluate its structural response. This discretization is carried out by the use of 'elements' that intend to represent such a

system, therefore the amount and type of elements that are needed to carry out a correct representation become a question. Another question is which modeling techniques best represent the system that is being analyzed. Other questions that arise during the modeling process might not be related to the validation, but are rather related to the design itself and whether some design features improve or hinder the overall performance of the system. The following questions arose when considering the icosahedral structure:

1. How many elements are needed in order to obtain accurate results?
2. At what thickness does the skin behaves like a membrane? In other words, when is the skin thickness thin enough to loss its bending stiffness?
3. What is the skin reaction to changes in material properties?
4. What material properties are needed such that skin deflection does not cause significant loss of buoyancy?
5. What BC are appropriate for the icosahedral model?
6. Considering the frame performance, which is better, hollow or solid beams?

These questions needed answers prior to considering the overall structure. Figure 20 shows the studies conducted to answer such questions. First, the finite element techniques were validated by comparing the square and circular membranes solutions, shown in Section 2.5 against the membrane and shell elements, latter discussed in detail. Second, a triangular model that represents the icosahedral skin was used to run convergence, thickness and material studies. Third, a frame model with an equivalent force method was used to run a convergence study, verify the effects of different BC and compare the

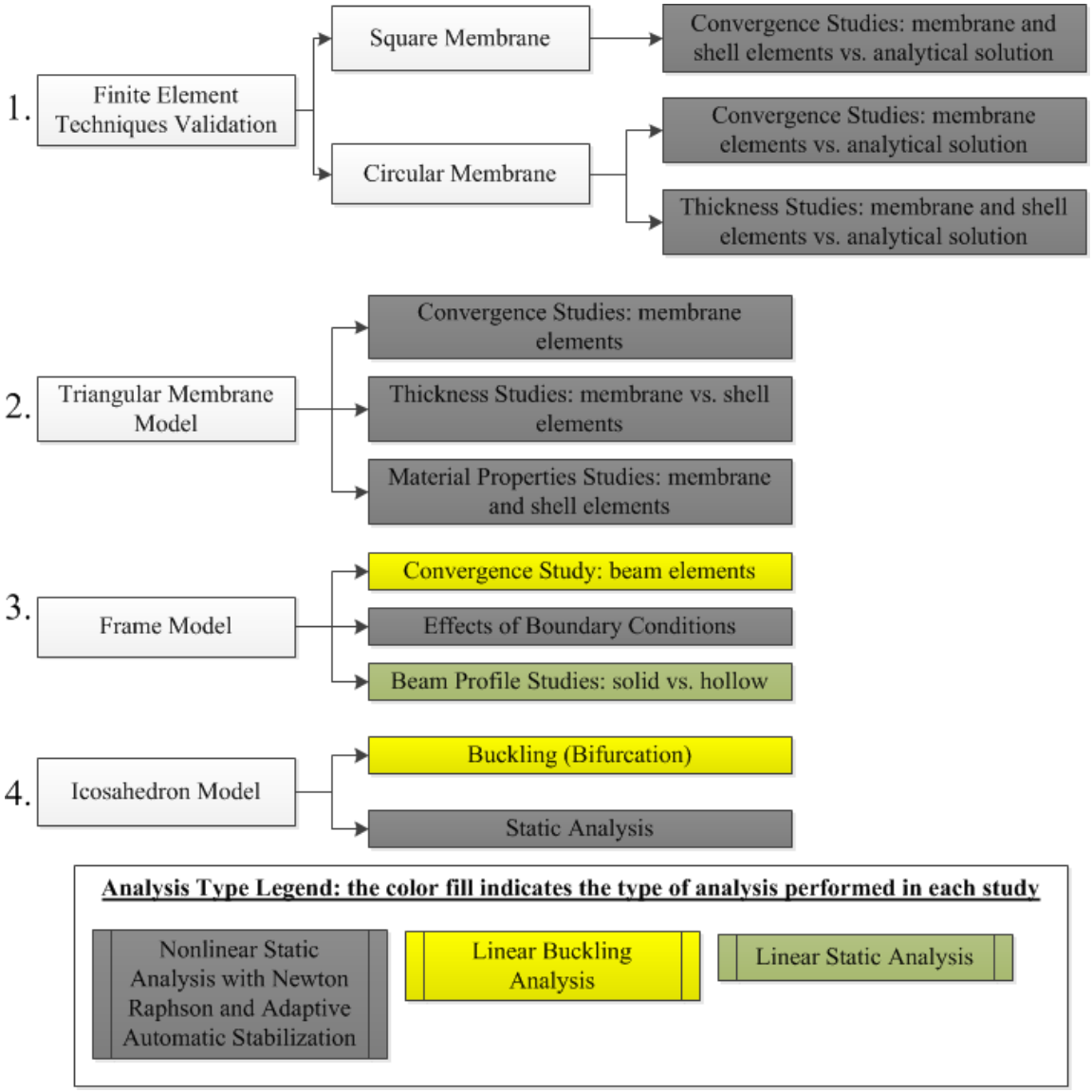


Figure 20: Modeling Studies. Shows an overview of the studies made prior to and with the icosahedral model in an orderly fashion.

effect of solid versus hollow beams. Finally, all the results were gathered and used to evaluate the icosahedral model.

This chapter starts off by discussing the process used to run all studies and analyses, then moves to a discussion of the dimensionality selection and the rationale behind it. Once that is established, the finite element techniques and their validation are presented. Afterward, the studies ran with triangle and frame models and their results are discussed. Finally, the conglomeration of techniques that are used to establish and evaluate the icosahedral structure is presented.

3.2 Process

The FEA was conducted in this research by using Abaqus [11] in combination with Matlab [34]. Abaqus itself provides three venues to analyze models: the Complete Abaqus Environment (CAE), the input file and the Python computing language [44][13]. Each model considered in this research was initially created using CAE, and the Python code was then extracted and modified to accommodate for changes in geometry, meshing characteristics, analysis type, BC, etc. Once the modified Python code was completed, Matlab was used to adapt, run and extract results from models. This process, shown in Figure 21, was repeated such that results could be compared.

As shown in Figure 21, the 'Main Routine' sends the FE settings, material inputs and geometric inputs to the subroutine through a counter that establishes the amount of analyses performed within it. Within the subroutine, the caller function takes the inputs

[13]Each of the modeling venues has its advantages and disadvantages. For example, the input file provides direct access to the FE processor without the need of creating the visual model, becoming advantageous for simple models that are already discretized and when conditions such as force magnitudes and boundaries change repeatedly, among others. On the other hand, CAE provides visual access to modules and a more guided process to create, analyze and view the model's results, but repetitive processes become time consuming and larger in storage size. Python provides access to aspects of both, it takes the same steps as with the CAE. But once it is created, the Python code can be modified to serve almost any purpose. Therefore, it becomes a great tool for repetitive processes where various modeling parameters can change.

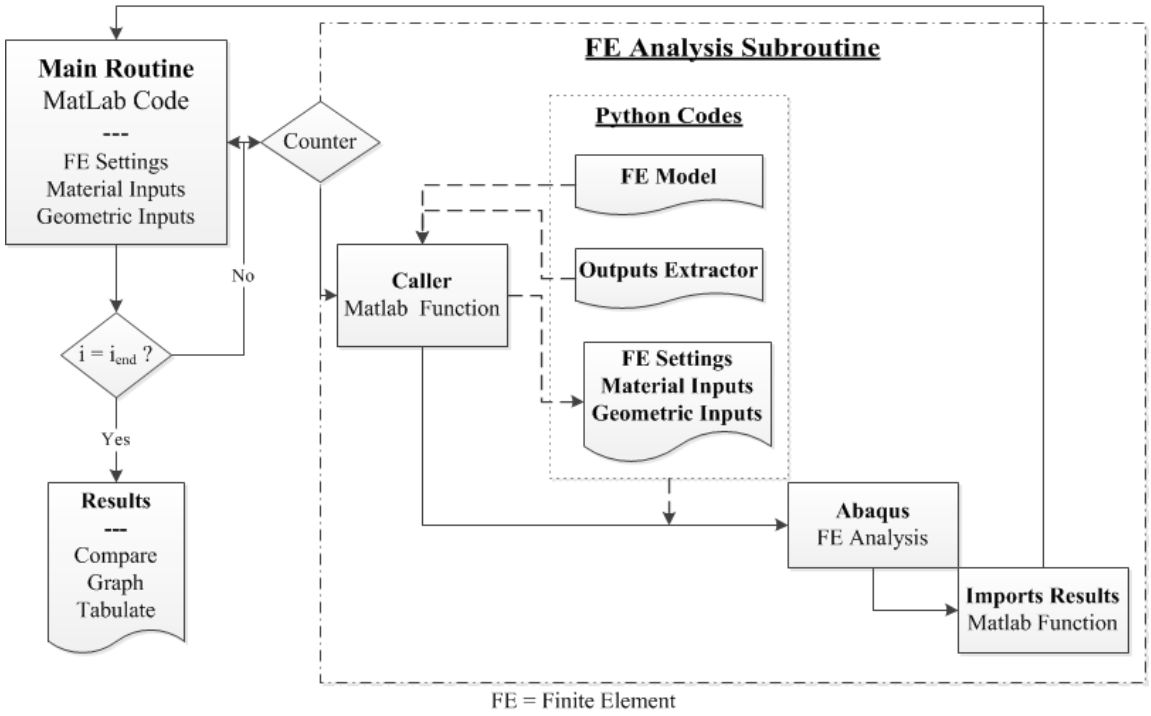


Figure 21: Models Analyses Flow Diagram. Inputs and settings are controlled in the main routine. Python codes establish each model, and the Matlab caller function sends those models to Abaqus for analysis. The results are then imported with Matlab for comparison.

and the FE settings from the main routine and creates a Python code with them, which are then sent along with the model and output extractor python codes to Abaqus for analysis. After Abaqus is done with the analysis, the results are read back with Matlab. This process is then repeated according to set specifications on the main routine. Finally, results are compared, graphed and/or tabulated in Matlab. Python codes for each of the models are included in Appendix B. Matlab routines and functions are included in Appendix C.

3.3 Dimensionality

The dimensionality of the icosahedron was one of the first considerations. The history of LTAV shows designs made to carry people and cargo, ergo they where large in size. History also shows that this type of vehicle suffered landing and maintenance problems, among others. Trying to move away from those problems, a small structure was considered; one that perhaps serves to carry small payloads and that was easy to transport and deploy. Therefore, the icosahedron dimensionality for all models considered in this thesis was chosen at that point to be 1 ft (0.3048 m) in diameter, measured from opposite vertices passing through the center.

3.4 Finite Element Techniques and Validation

This section introduces the first set of studies conducted, as shown in Figure 20. First, the type of elements and the analysis techniques are described. Then, the studies related to the square and circular membranes are presented.

3.4.1 Elements.

Three types of elements are considered: (1) the beam element, B32; (2) the membrane element, M3D3; and (3) the shell element, S3R. The beam element is used to represent the icosahedral frame members; the membrane and shell elements are used to represent the icosahedral skin. For all of them, the element coding is established by Abaqus.

Beam theory allows us to approximate the behavior of a slender structural component, such as the frame members, by reducing it dimensionally from the 'true' three-dimensional to a one-dimensional behavior. The main benefit of using such approximation is that beam elements are geometrically simple with less DOF, compared

to three-dimensional elements, which reduces the computing power needed for analysis. Although, this approximation relies on the assumption that the deformation can be estimated from variables that are functions of position along the beam axis only. The B32 element, specifically, is a quadratic element that is used in a three-dimensional space, based on Timoshenko's beam theory. A quadratic beam element is composed of three nodes with six DOF at each node, three translational and three rotational, therefore is capable of capturing the effects of both axial and transverse loads. The main difference betweent Timoshenko's and Euler-Bernoulli's beam theories is that Timoshenko's include transverse shear deformation, which is the capacity of capturing in-plane deformation caused by the beam's bending moment. The B32 element is used for all the analyses involving the frame. See Section 28.3 of Reference 12 for more details.

Considering the skin, two type of elements are compared: a membrane element, M3D3, and a shell element, S3R. The analytical and FE definition of what a shell and a membrane are is the same, with the exception that both element types can be applied to either flat or curved surfaces. In other words, the shell element carries both membrane and bending stiffnesses and can be used for both initially flat and curved surfaces [12, Ch. 28.6], while the membrane only carries membrane stiffness but can still be used for both initially flat and curved surfaces [12, Ch. 28.1].

The M3D3 is a three-dimensional triangular surface element with three nodes, in which each node has three displacement DOF. This element is commonly used to represent thin surfaces with no bending stiffness, therefore has no rotational DOF [12, Ch. 28.1.1]. The S3R is also a three dimensional triangular surface element, but it has all six DOF such that it carries both membrane and bending stiffnesses, with finite

membrane strains. A pictographic representation of the triangular element, as well as the DOF for both shell and membrane elements, are shown in Figure 22. In regards to the DOF numbering system: the displacement DOF are 1, 2 and 3 in the x, y, and z directions, respectively, and rotational DOF are 4, 5 and 6 about the x, y, and z directions, respectively. The main reason of using triangular elements over square elements is that the formers allow for an homogeneous mesh in each icosahedral face.

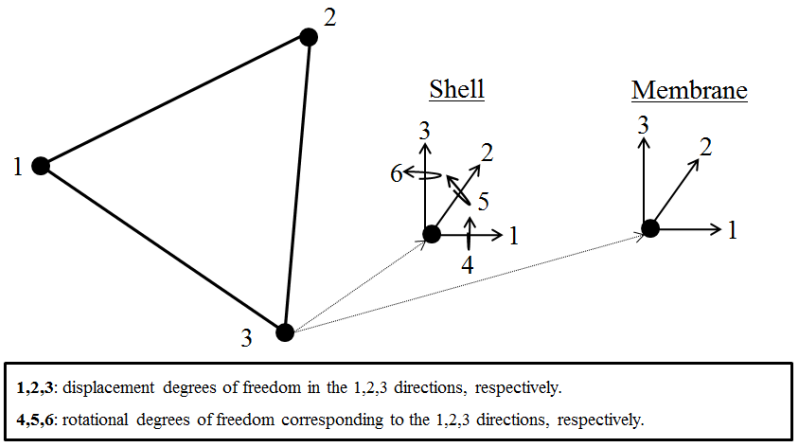


Figure 22: Triangular Surface Element Representation

Additionally, the S3R is a hybrid element that uses thin shell theory and transitions to thick shell theory as thickness increases, making it a general purpose element. The 'R' stands for reduced integration, which uses a lower order integration to form the element stiffness. One of the benefits of reduced integration is the use of less integration points, resulting in less computing time and storage space. Another benefit is the accuracy of results. The strain and stress in reduced integration elements are calculated at Barlow points, which provide optimal accuracy [8]. This sometimes can be comparatively observed in large displacement analyses with transverse loading where fully integrated

elements show overly stiff behavior associated with shear-locking, while reduced integration elements show relaxed and improved performance [30]. The drawback of using reduced integration is that it can introduce zero-energy deformation modes that produce zero strain and stresses, called hourglass modes, leading to inaccurate results. Abaqus controls hourglass modes by adding a small artificial stiffness to zero-energy modes [13, Ch. 3.1.1]. Both elements are used in convergence studies related to the skin, mainly to locate the transition point at which the shell element shows membrane behavior as a result of its thinness.

3.4.2 Analysis Techniques.

Two aspects are considered when selecting the analysis technique to evaluate the various models used in this thesis. First, large displacement analysis is inherently a nonlinear problem. Abaqus has two techniques to solve nonlinear static problems: Riks and Newton Raphson; the latter is used in this thesis. A one-dimensional description of the Newton Raphson technique is presented in Section 2.4.2. The main difference between them is that a load (in load controlled analyses) or displacement (in displacement controlled analyses) input is required for the Newton Raphson, while the Riks solves simultaneously for load and displacement. Unlike the Newton Raphson, the Riks technique has the capacity to follow solution paths where snap-through and snap-back occurs, capturing buckling and post-buckling behavior of a structure. These types of global instabilities are well managed with Riks, but instabilities that cause local transfer of strain energy from one part to the other within the model might cause convergence issues. On the other hand, the Newton Raphson technique has the capacity of adding adaptive automatic stabilization to equilibrium equations. Stabilization adds

viscous forces on the form $F_v = cM^*v$ to overcome those local instabilities, where M^* is an artificial mass matrix calculated with unit density, c is the damping factor and v is the nodal velocities vector. The 'automatic' feature adds volume-proportional damping and the 'adaptive' feature varies the damping factor spatially and/or with time, controlled by the convergence history and the ratio of dissipated energy to total strain energy. This allows for a converged solution while minimizing the effect introduced by damping. The maximum ratio of dissipated energy to total strain energy is set by default and left at 0.05 [12, Ch. 7.1.1].

The second aspect considered in the analysis selection is the membrane behavior. Initially flat and stress free membranes have no stiffness; therefore out-of-plane loading, such as pressure, causes numerical singularities and convergence difficulties. One option is to pre-stress the membrane such that it can acquire stiffness. Another option is the use of stabilization, such that for the first increment where the membrane has no stiffness, the viscous forces eliminate the singularities and once some out-of-plane deformation has developed, the membrane acquires stiffness, resisting out-of-plane loading. Therefore, the static step with adaptive automatic stabilization for a maximum ratio of dissipated energy to total strain energy of 0.05 is used for all nonlinear analyses conducted in this thesis.

Linear static and linear buckling analyses are also conducted in this thesis. The linear static analysis is a procedure that solves for the equilibrium of a structure given the applied loads assuming there is no stiffness changes, therefore solving for displacements without the need of an iterative process. This procedure is only used to compare the relative behavior of hollow beam profiles versus the solid profile when considering the frame standalone. The linear buckling analysis is a perturbation procedure that estimates

the eigenvalues and vectors that represent the critical (bifurcation) loads and the mode shapes corresponding to each critical load, respectively. This procedure is used to estimate the mode shapes of the icosahedral structure, as well as as to evaluate the effect that hollow beam profiles have in the critical load.

3.4.3 Square Membrane.

Referring back to Figure 20, the first step of the modeling process is to validate the proposed FE techniques in order to properly use them in the icosahedral structure. Three elements have been described: B32 (beam), S3R (shell) and M3D3 (membrane), along with several analysis techniques, including the Newton Raphson with adaptive automatic stabilization. Additionally, two analytical solutions for the square membrane are presented in Section 2.5 by Equation (2.32) to Equation (2.35). These solutions are used to run convergence studies that not only validate the analysis techniques, but also provide the correct number of elements needed in order to achieve an accurate solution for the square membrane, considering both S3R and M3D3 elements.

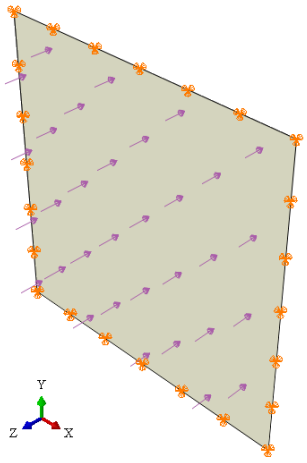


Figure 23: Square Membrane Model. Arrows represent the surface pressure. Orange symbols represent the fixed displacement boundary conditions in the x,y, and z axes.

The square membrane model is composed on a flat surface with the displacements DOF tied around the edges and SL pressure load applied to and parallel to the entire surface. The model is shown in Figure 23; the symbols at the edges represent the BC and

Table 2: Square Membrane Model Properties

Area **(*m*^{2})**	0.0111
Dimensions **(*m*)**	l_{edge} = 0.1054
Thickness **(*m*)**	5e-5
Load **(*Pa*)**	101,325
Boundary Conditions	U1=U2=U3=0 along edges
Analysis	Static Nonlinear with Adaptive Automatic Stabilization
Modulus of Elasticity **(*GPa*)**	303
Poison's ratio	0.3
Density **(*kg*/*m*^{3})**	1560
Element Type	M3D3/S3R

the arrows represent the pressure load. The BC were selected in agreement with analytical solutions; having the rotational DOF free extends from the fact that the membrane has no bending stiffness,since it is carried through rotations. Using the SL pressure as the magnitude extends from the fact that an LTAV under a vacuum is subjected to no more than the pressure at SL (details are discussed in Section 2.3). The model has the same surface area as one triangular face of the icosahedral skin. Additionally, it was discretized by selecting the amount of elements desired per edge, called 'seeding', where all edges shared the same seeding number and element size, producing a homogeneous mesh. Model properties are listed in Table 2.

The purpose of considering both elements is to evaluate their behavior against analytical solutions and to confirm that the bending stiffness diminishes in the shell

element for small thicknesses, validating the use of BC with free rotations. For this study, the membrane is assigned a thickness of 0.05 mm, thin enough to expect membrane behavior out of the shell element. The study was conducted by varying the edge seeding from 8 to 43 seeds, in increments of 1, resulting in 128 to 3698 elements, respectively. In order to compare the FE solution with analytical solutions, the center displacement is selected as the delineating factor for convergence. Results obtained from analytical solutions demonstrate that they predict the center displacement within 10% of each other. It was found that the FE solution agrees very well with Seide's solution, while Timoshenko's solution remains within 10% of the rest. Element's center displacement magnitude and % error vs number (#) are shown in Figure 24, for both FE and Seides solutions.

Note that both elements agree very well with each other for every number of elements. Also, results show that 390 elements are enough for convergence within 1%, using either element. There is significant oscillation in both elements as a result of the numerical approximation, but it diminishes as the number of elements increases. Results obtained from all solutions are tabulated in Section A.1. The Python code that produces the square membrane is included in Section B.2.

At this point, FE techniques has been validated for a rectilinear configuration, for both shell and membrane elements. But the question regarding the effect of thickness on the solution is still unanswered.

3.4.4 Circular Membrane.

Two studies were conducted for the circular membrane: a convergence study and a thickness study. As with the square membrane, the purpose of the convergence study is to

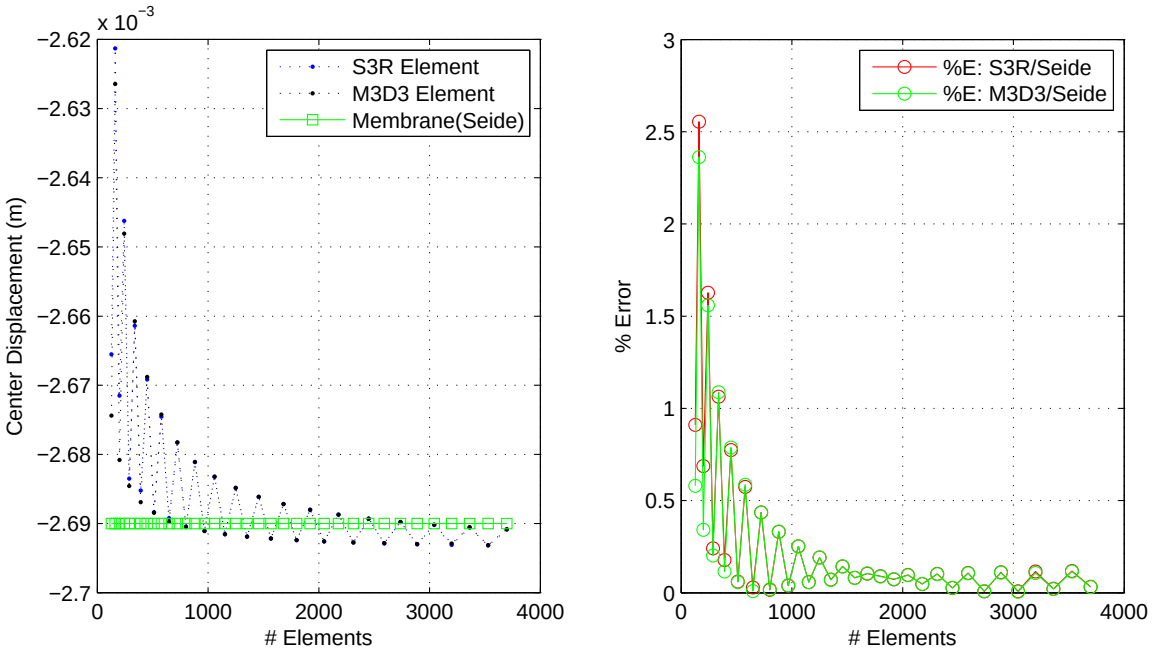


Figure 24: Square Membrane Convergence Study Results: Center Displacement and % Error versus # of Elements

validate the FE techniques and find the discretization that allows an accurate solution. On the other hand, the purpose of the thickness study is to find the thickness point at which the shell starts behaving like a membrane. In this case, that point in not particularly important for the analysis of the icosahedron, but confirms that the point can be found within reasonable thicknesses.

The circular membrane model features are similar to those of the square membrane model. It is composed on a flat surface with the displacements DOF tied around the edge and SL pressure load applied to and parallel to the entire surface. The circular membrane model is shown in Figure 25; symbols at the edge represent the BC and arrows represent the pressure load. This model also has the same surface area as a triangular face of the

icosahedral skin. Additionally, it was discretized by seeding the edge with the same element size. Model properties are shown in Table 3.

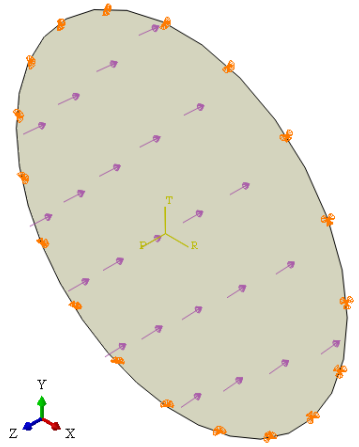


Figure 25: Circular Membrane Model. Arrows represent the surface pressure. Orange symbols represent the fixed displacement boundary conditions in the x,y, and z axes.

The circular membrane convergence study consisted of discretizing the mesh made of M3D3 elements, from 5 to 51 edge seeds, in increments of 1, representing 5 to 475 elements, respectively. For each analysis run, the center displacement and von Mises stress were compared against the analytical solution, provided by Equation (2.29) and Equation (2.30). Convergence study results are shown in Figure 26. Note that for more that 50 elements, the error is less that 5% for both the displacement and stress. Furthermore, results tabulated in Section A.2 show that 172 elements are sufficient to achieve convergence within 1% for both displacement and stress. While stress increases exponentially with the number (#) of elements, center displacement shows periodic behavior that damps out as the number of elements increases.

Table 3: Circular Membrane Model Properties

Area (m^2)	0.0111
Dimensions (m)	*diameter* = 0.119
Thickness (m)	1e-3
Load (Pa)	101,325
Boundary Conditions	U1=U2=U3=0 along edges
Analysis	Static Nonlinear with Adaptive Automatic Stabilization
Modulus of Elasticity (GPa)	303
Poison's ratio	0.3
Density (kg/m^3)	1560
Element Type	M3D3/S3R

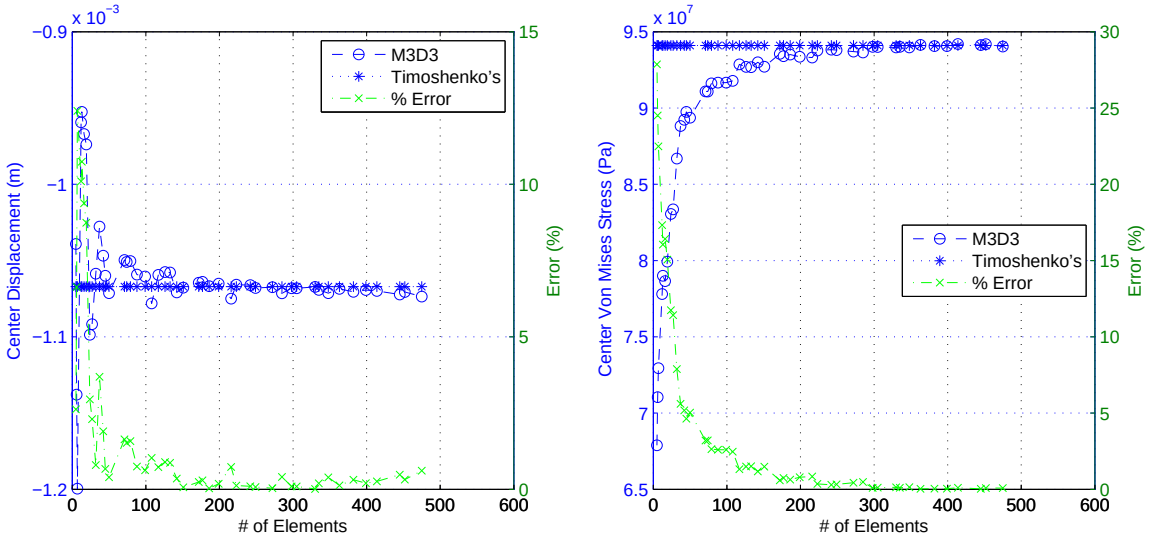


Figure 26: Circular Membrane Convergence Study Results: Center Displacement and Stress versus # of Elements

The thickness convergence study comparing the analytical solution against the shell (S3R) element for thicknesses from $5.0x10^{-6}$ to $2.0x10^{-1}$ (m). The model had the same properties listed in Table 3, but the mesh was fixed at 452 elements and different thicknesses. One of the main characteristics of the shell element is that even at small thickness, it carries bending, therefore a cross-sectional stress distribution is expected (see Figure 10 in Section 2.5). Ideally, there is a thickness point at which that stress distribution approaches a constant value across the entire thickness, as the membrane does. Center out-of-plane displacement and stress versus thickness are shown in Figure 27, for both the shell element and the analytical solution. Results show a fairly constant % error in both displacement and stress for thicknesses less than 0.5 mm. But there is a significant jump in % error for thicknesses less than 0.05 mm. Furthermore, results after that point have a significant change in slope, suggesting that bending stiffness became significant. The tabulated results, included in Section A.3, indicate that a thickness less than 0.7 mm produces solutions agreement within 5%. The stress analytical solution is compared against the shell stress at the mid-plane. The Python code that produces the circular membrane is included in Section B.1.

At this point, the FE techniques, specifically the use of the Newton Raphson technique and both membrane and shell elements to model the behavior of initially flat membranes, have been validated with both square and circular models. In order to evaluate the membrane behavior of the icosahedral skin, a triangular model is considered next.

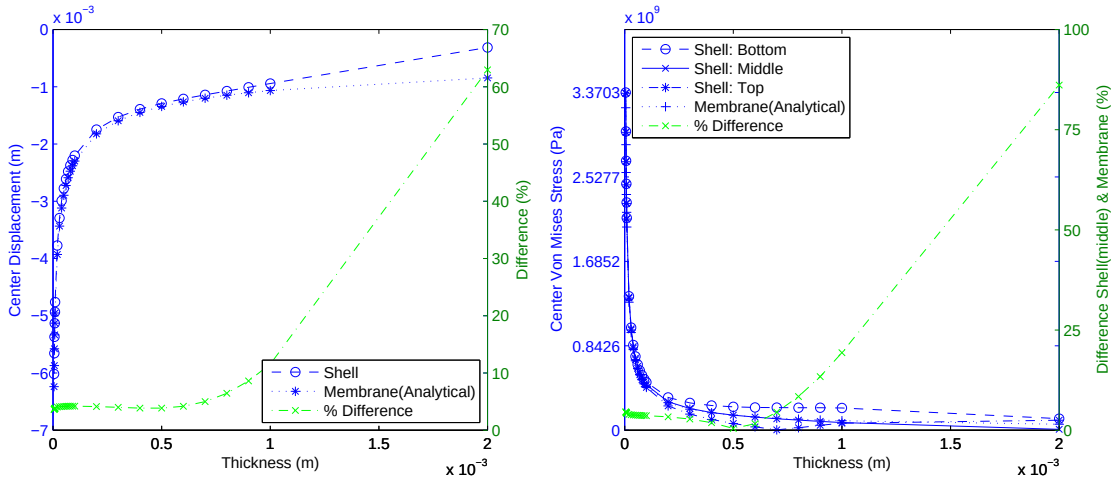


Figure 27: Circular Membrane Thickness Study Results: Center Displacement and Stress versus Thickness

3.5 Triangular Membrane

In order to answer the first four questions formulated on Section 3.1, a triangular model that represents the icosahedral skin is considered. The model consists of an equilateral triangular surface of the same area as a triangle of the icosahedral skin, based on a icosahedron with a diameter of 0.3048 m (1 ft.). The model is shown in Figure 28; symbols at the edges represent the BC and the arrows represent the pressure load. This model assumes that the frame remains rigid during deformation, therefore all three edges have fixed displacement DOF. Nonetheless, the rotational

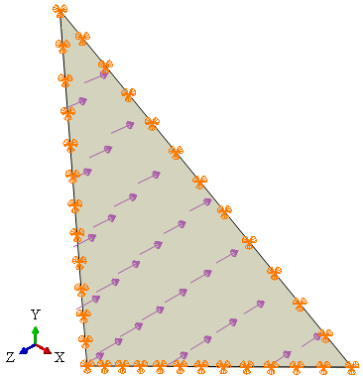


Figure 28: Triangular Membrane Model. Arrows represent the constant pressure applied to the surface and orange symbols represent the fixed displacement boundary conditions in the x,y, and z axes.

Table 4: Triangular Membrane Model Properties (unless otherwise stated)

Area (m^2)	0.0111
Dimensions (m)	l_{edge} = 0.1602
Thickness (m)	1e-3
Load (Pa)	101,325
Boundary Conditions	U1=U2=U3=0 along edges
Analysis	Static Nonlinear with Adaptive Automatic Stabilization
Modulus of Elasticity (GPa)	303
Poison's ratio	0.25
Density (kg/m^3)	1560
Element Type	M3D3/S3R

DOF remain free due to the expected membrane behavior. As with previous models, the SL pressure is used throughout the surface. Model properties are listed in Table 4. The Newton Raphson with adaptive automatic stabilization technique is used for all analyses involving the triangular model.

From the W/B point of view, having these boundary conditions limits the amount of volume loss since the skin deflection around the edges is eliminated. From the structural point of view, using such model can underestimate the effect that the frame has on the skin since former, when connected to the latter, will not remain rigid. On the other hand, using such a model provides a venue to efficiently estimate skin behavior. The Python code that produces the triangular membrane is included in Section B.3.

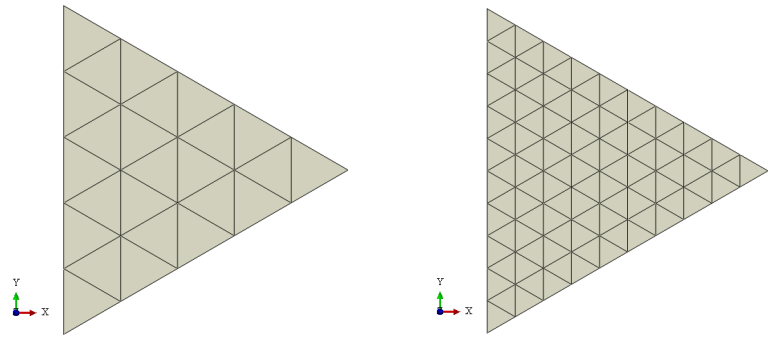
First, a convergence study is conducted to find the discretization needed in order to achieve a converged solution with the membrane element. Second, a thickness study is

conducted to find the point at which the shell element behaves like a membrane. Finally, a material properties study is conducted to understand changes in membrane's behavior due to variations in material properties. Is important to clarify that, since no analytical solution was found for the triangular membrane, studies conducted for this thesis compare the relative performance of the elements considered.

3.5.1 Convergence Study.

This convergence study was conducted by seeding homogeneously the edges of the membrane (M3D3) element (shown in Figure 22), such that each edge has the same amount of elements. This allows for a mesh composed of elements of the same size. Seeds along the edges are shown in Figure 29, for both 5 and 10 seeds per edge. Note that all edges have the same amount of triangles adjacent to them.

The study was conducted in increments of one, from 5 to 50 seeds per edge, representing 25 to 2296 elements, respectively; for a total of 46 analyses. The displacement and stress at the center were tracked for convergence by using two methods. First, the % difference was calculated by considering the i seeding value against the $i-1$ value. Second, the % difference was calculated by considering the i seeding value against the last, i_{end}, value. Results are shown in Figure 30. Note that while the displacement converges very steadily, the stress has more variation and a higher % difference. Nonetheless, results indicate that 324 elements (18 seeds per edge) are sufficient for displacement and stress convergence within 1% and 5%, respectively, compared to the 50 seeds case. Results are tabulated in Section A.4.



(a) 5 Seeds per Edge (b) 10 Seeds per Edge

Figure 29: Triangular Membrane Mesh Comparison

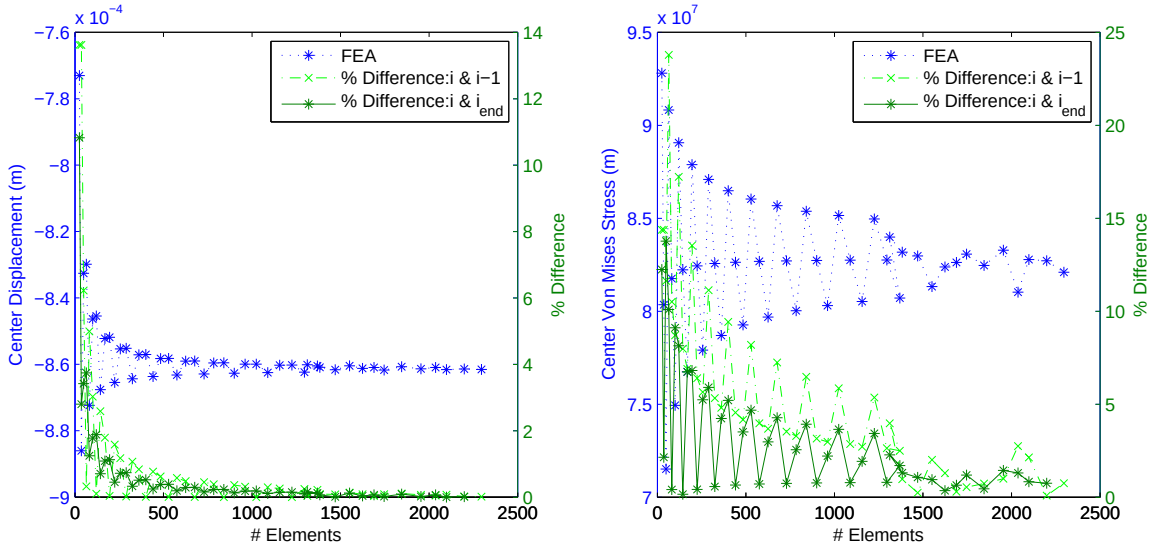


Figure 30: Triangular Membrane Convergence Study Results: Center Displacement and Stress versus # of Elements

3.5.2 Thickness Study.

As with the thickness study of the circular membrane, the purpose of this study is to find the thickness point at which the shell element behaves like a membrane. Since there is no analytical solution in this case, the membrane element is used for comparison. The

model is discretized with 1296 elements (50 seeds), which is above the point of convergence. Quantities considered are center displacement, von Mises stress and strain energy. In case of the shell element, the top, middle and bottom stress are considered. Strain energy provides a globalized measure that relates both stress and strain results of the entire model, contrary to displacement and stress which are localized values.

The study yielded results similar to those of the thickness study conducted for the circular membrane (see Figure 27), therefore plots are not presented in this section. Tabulated results, included in Section A.5, indicate that a thickness $t_{skin} \leq 0.02\ mm$ provides a shell and membrane agreement within 1%. Therefore, the skin thickness on further analyses will be verified to identify if this point is being crossed.

3.5.3 Material Properties Study.

The purpose of the material properties study is to approximate the icosahedral skin response due to material properties changes, where the skin is represented by the triangle. The ideal W/B is set at 0.4 (see Equation (2.14)) and the skin thickness results from the given density and set W/B (see Equation (2.15)). This study was performed with 400 elements, for both M3D3 and S3R elements, but since they provided almost identical results, only M3D3 results are presented. To analyze changes, a three-dimensional input space was created with the following parameters:

$$412 \leq \rho \leq 3000 \quad kg/m^3$$
$$100 \leq E \leq 1000 \quad GPa$$
$$0.1 \leq \nu \leq 0.4$$

where:

ρ = density; E = modulus of elasticity; ν = Poison's ratio

Vectors of length 25 were created for ρ and E, where initial and final values of each vector is given by the limits provided. On the other hand, ν was changed by 0.1 increments within the provided limits. A total of 2500 analyses were done based on the three dimensional design space created by their combinations. The density range was selected such that the maximum skin thickness did not exceed 0.2 mm (the maximum value at which the shell element behaves as a membrane element). Given each input variable combination, the following quantities were considered: center out-of-plane displacement, strain energy and skin W/B after deformation. Center displacement, important to quantify maximum displacement, provides a local or node dependent result. On the hand, strain energy provides a globalized measure that relates both stress and strain results of the entire model.

The skin W/B after deformation was calculated by including an estimation of the volume lost due to triangle's deflection. To estimate the volume loss, the deflected surface is integrated numerically using Matlab functions' 'quad2d' [48] with 'gridddata'; the latter fits the triangular surface given by the deflected nodes. The integration limits are given by Equation (3.1), as a result of the established geometry shown in Figure 31.

$$
\begin{aligned}
-\frac{h}{3} \leq x \leq \frac{2h}{3} \\
-\frac{2h}{3\sqrt{3}} + \frac{x}{\sqrt{3}} \leq y \leq \frac{2h}{3\sqrt{3}} - \frac{x}{\sqrt{3}}
\end{aligned}
\tag{3.1}
$$

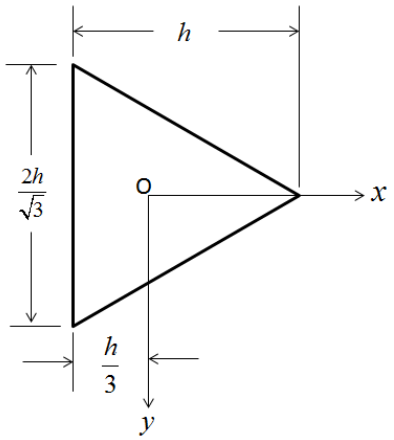
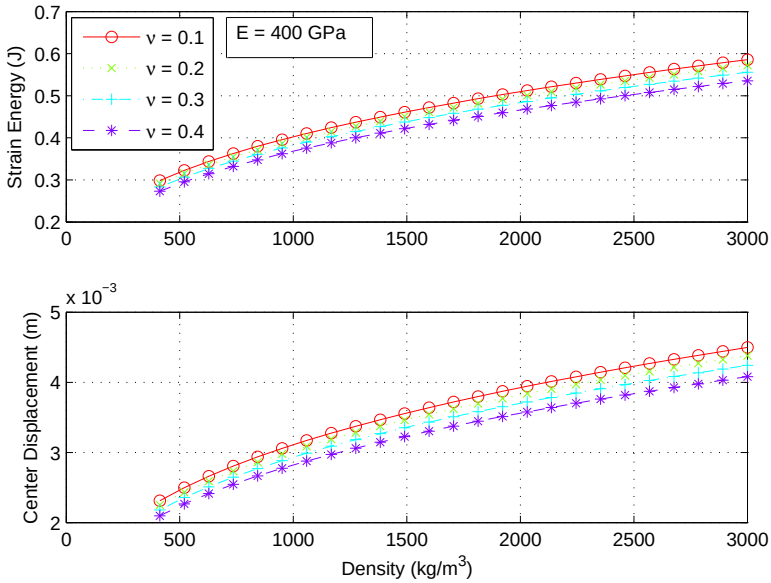
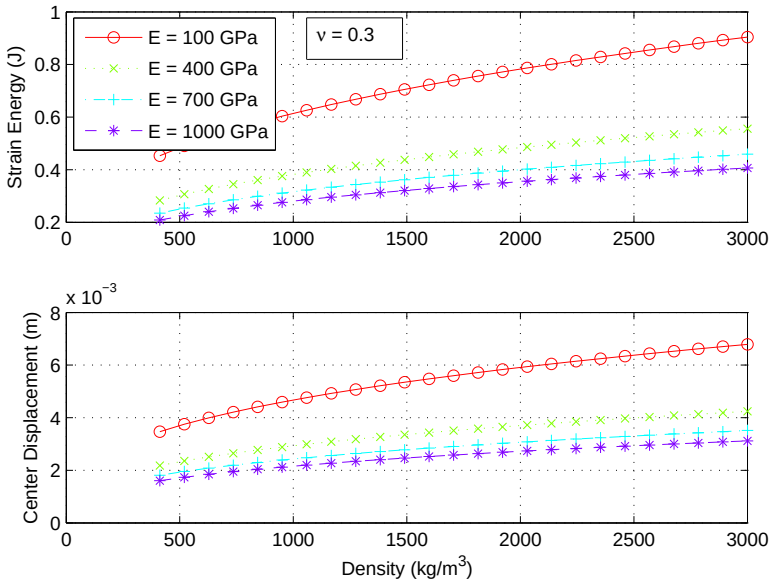


Figure 31: Triangular Membrane Geometry. *O* represents the center and *h* represents the height.

Displacement and strain energy curves are shown in Figure 32 for constant values of modulus of elasticity and Poison's ratio. Ideally, less displacement and strain energy is better. Changes in modulus of elasticity cause significant response in the triangle (Figure 32a). Furthermore, changes can be as high as 73% between the limits considered, as shown by tabulated results in Section A.6. On the contrary, there is little variation between Poisons ratios (Figure 32b); in fact, high Poison's ratios lead to stiffer responses by no more than 10%. Another consideration is the material's density. Low material density leads to high skin thickness, as shown by Equation (2.15). It can be shown that the response can change up to 64% within density limits. Additionally, strain energy and center displacement are compared against modulus of elasticity for fixed Poisson's ratio and density in Figure 33. Note that the response changes considerably for low moduli, suggesting that there is a trade space. Nonetheless, the final W/B (including the volume reduction) shows the following range: $0.41 \leq W/B_{skin} \leq 0.44$ within the design space considered, therefore not graphed here. Details are discussed in Section A.6. Note that regardless of the material properties selected, large displacements are observed. From a numerical point of view, this can become an issue, but the use of stabilization in the Newton Raphson technique allowed for a smooth convergence. The analyses showed sensitivity to the selection of the initial load increment. This was managed in Matlab by automatically adjusting the initial load increment every time convergence issues arose and rerunning those analyses.



(a) Constant Modulus of Elasticity



(b) Constant Poison's Ratio

Figure 32: Triangular Membrane Material Properties Study Results: Center Displacement and Strain Energy versus Density

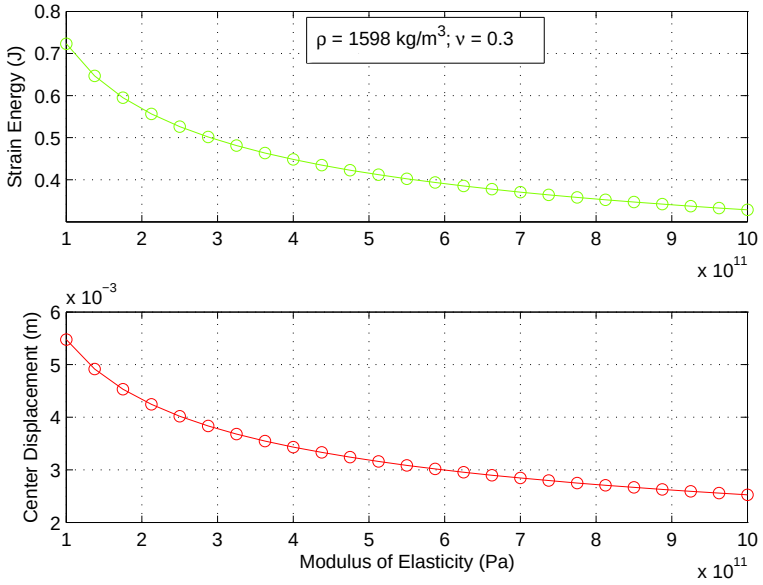


Figure 33: Triangular Membrane Material Properties Study Results: Center Displacement and Strain Energy versus Modulus of Elasticity

At this point, the model discretization and the expected skin behavior are known, including the effect of thickness, density, Poison's ratio and modulus of elasticity. Furthermore, it was found that changes in material properties have minimum effect in the skin W/B.

3.6 Frame Standalone

The icosahedral frame standalone model provided a decision guide for the geometric characteristics and the proper BC. Three studies were conducted: a convergence study, a beam profile study and a BC study. Prior to conducting such studies, the geometric definition of the icosahedral structure was established. Additionally, an important

question was considered: how can the pressure applied to the icosahedral skin be modeled in the frame? This question was answered with the use of the coupling constraint.

3.6.1 Geometric Definition.

The coordinates of each of the 12 icosahedral vertices were obtained by using the Matlab function: icosahedron_coordinates.m (provided in Section C.2). This function was created with a coding provided by T.T. Metlen [38, 141]. Since each of vertices lies on an imaginary circumscribed sphere, the location of each vertex is established using of spherical coordinates and then transformed into Cartesian coordinates. The spherical coordinate system is defined as shown in Figure 34, where θ is an angle measured from the x axis to the vector OP, ϕ is an angle measured counterclockwise from the xy plane to the vector OP, and r is the length of vector OP such that $OP = OP(r, \theta, \phi)$; with transformation into the Cartesian coordinate system: $x = r\ sin\phi\ cos\theta$, $y = r\ cos\phi\ sin\theta$ and $z = r\ sin\phi$. Placing the icosahedral center at $(0, 0, 0)$, top and bottom vertices can be taken as the north and south poles, defined at $(r, \pm 90°, 0)$, for any r value (see Figure 5). Of the 10 vertices left, five are located at the upper hemisphere equally spaced by $\theta = 72°$ at a constant $\phi = 26.6°$, and the other five at the lower hemisphere equally spaced by $\theta = 72°$ at a constant $\phi = -26.6°$. Once the icosahedron radius, r, is established, the spherical coordinates are transformed to Cartesian. See Reference 38 for more details. Note that in order to

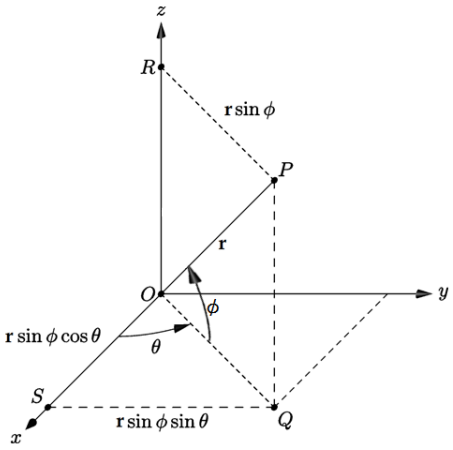


Figure 34: Spherical to Cartesian Coordinates Systems Transformation [5]

calculate the volume (Equation (2.5) to Equation (2.10)), a geometric approach was taken, rather than a trigonometric one, but both yielded the same angular relationships.

3.6.2 Coupling Constraint.

The coupling constraint is a surface based constraint provided by Abaqus that couples the motion of a surface's collection of nodes to the motion of a surface's reference point [12, Ch. 33.3.2]. Once the model is discretized, the reference point becomes a node, allowing for the surface's mesh to couple with the reference point; the surface's coupled nodes are referred as 'coupling nodes'. Abaqus offers various types of coupling constraints, including the distributed coupling. In general, distributed coupling constraints the rotation and translation of the reference node to the coupling nodes. It transmits loads and BC applied to the reference node through the use of weight factors at the coupling nodes [37]. It distributes loads such that the resulting forces (and moments) at the coupling nodes are equivalent to forces (and moments) at the reference node. The rotational DOF can be released from the constraint, allowing the transfer of forces, but not of moments. The default weighting method sets all weight factors to 1, but linear, quadratic and cubic weight factors can be implemented if desired. Additionally, this coupling constraint is available for both geometrically linear and nonlinear analyses.

Considering the use of such constraint to model the pressure transfer from the icosahedral skin to the frame, the equivalent pressure at each triangle can be given by: $F = PA$, where A is the triangle's area, P is the pressure, and F is the equivalent load. Before using the constraint in the frame, a triangular model with the equivalent load applied to a reference point located at the center is created (shown in Figure 35) and compared to the pressure model (shown in Figure 28).

After discretizing the model with 400 M3D3 membrane elements, displacement and von Mises stress are compared by using nonlinear analysis in both pressure and equivalent load models. Errors are calculated with respect to the pressure model by comparing the displacement nodal and stress elemental results. For displacement, mean and maximum errors are 0.3% and 0.5%, respectively. For stress, mean and maximum errors are 1.5% and 83.2%, respectively. Displacement and stress contours are shown in Figure 36 and Figure 37, respectively. While displacement contours show clear similarity, stress contours deviate close to the center, suggesting that the constraint causes a change in the stiffness matrix. Note that while pressure is a follower force, the equivalent force applied to the reference node will always remain perpendicular to the initial configuration, therefore causing a change in the membrane stiffness (see Section 2.4.1 for more details).

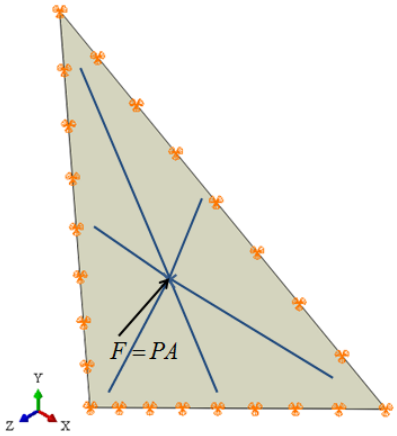
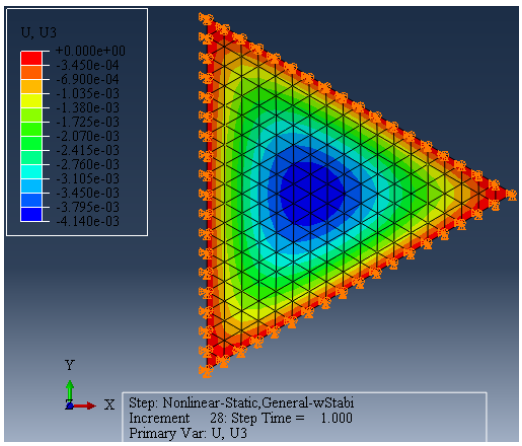
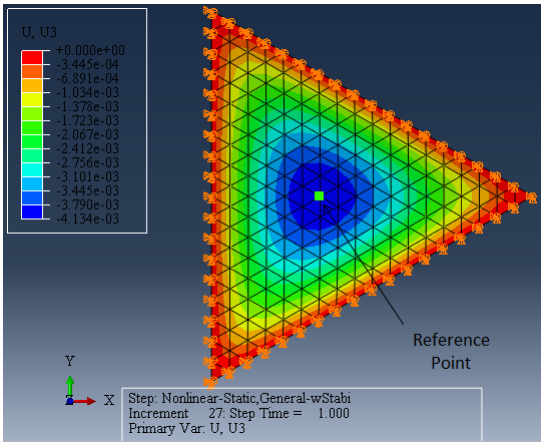


Figure 35: Triangular Membrane Model with Coupling Constraint. A is the triangle's area, P is the pressure, and F is the equivalent load. The coupling constraint is represented by the blue lines.

Regardless of the difference in stress that the coupling constraint showed in the triangle, it provides a method to estimate the forces transferred to the frame. This estimation relies on the assumption that all the pressure magnitude is transferred to the frame and that the skin provides no stiffness assistance to the frame. Assuming that the skin behaves as a membrane, just the displacement DOF are constrained, such that only

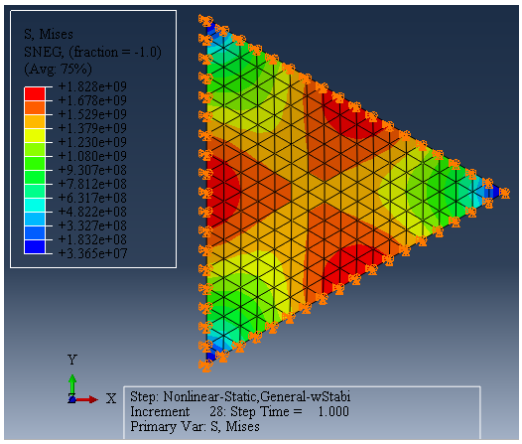


(a) Pressure Model

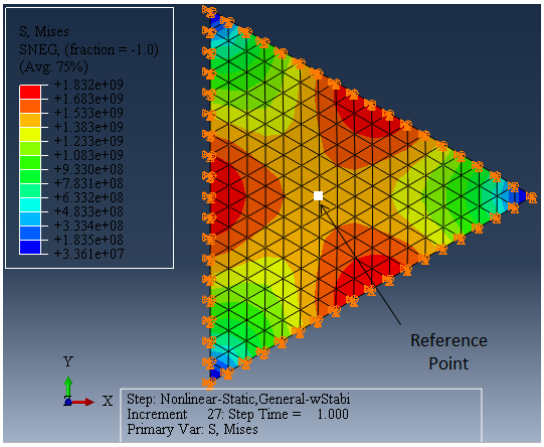


(b) Coupling Model

Figure 36: Triangular Membrane Coupling Constraint Validation: Out-of-Plane Displacement Contours



(a) Pressure Model



(b) Coupling Model

Figure 37: Triangular Membrane Coupling Constraint Validation: Out-of-Plane von Mises Stress Contours

forces (not moments) are distributed to the frame. Using a spherical coordinate system (refer to Figure 34), the equivalent force is applied to reference points located at the

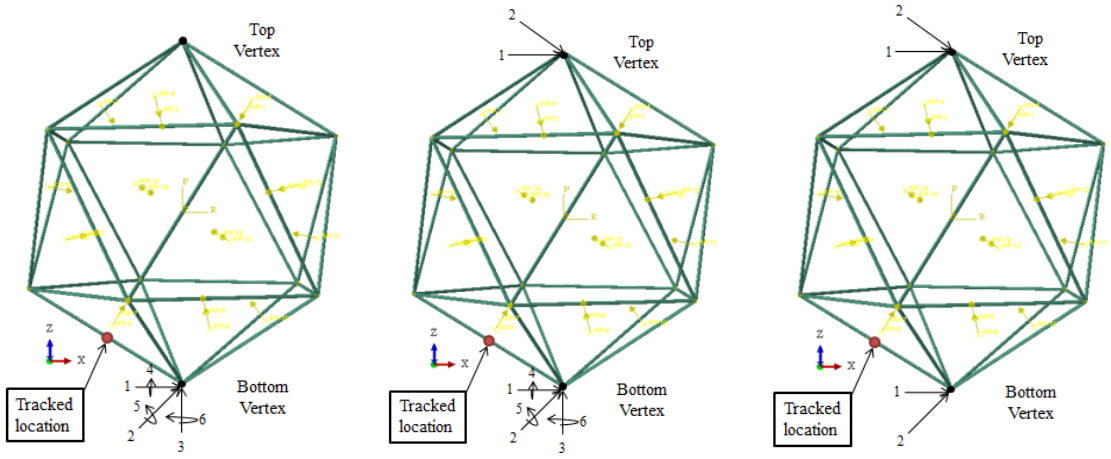
center of each triangle created by the 30 frame members in the radial direction (towards the center). Each reference point is coupled via separate coupling constraints to the respective beams. This model is used for subsequent frame studies to discretize the frame and evaluate the effects of different boundary conditions and beam profiles.

3.6.3 Boundary Conditions Study.

One of the main features of the icosahedron is symmetry. This property provides several structural advantages such as equal surface loading, improved stress distribution and buckling retardation. Additionally, the actual design will have no BC once afloat. The FEA requires for the model to have BC since otherwise the static analysis runs into singularities. Therefore, is important to select them such that symmetry is maintained throughout the analysis.

Three BC are considered, as shown in Figure 38. The first has the bottom vertex fixed Figure 38a), therefore all six DOF are constrained. The second has the bottom vertex fixed and the top vertex with the DOF 1 and 2 constrained (Figure 38b). The third has both bottom and top vertices with only DOF 1 and 2 constrained (Figure 38c).

The frame model is discretized with 1062 B32 beam elements and an initial W/B of 0.35. Beams are hollow with a beam thickness to radius of 0.05. Using nonlinear analysis, each of the BC are analyzed and compared. Displacement contours for all three acBC are shown in Figure 39. In the first BC, non-symmetric behavior is clearly shown close to the bottom vertex (Figure 39a). Behavior in the second BC improved, but non-symmetry is still seen around the bottom vertex (Figure 39a). At this point, it can be deducted that having the bottom vertex fixed is causing the unsymmetrical response. The third BC is shown in Figure 39a. In this case, the icosahedron shows symmetrical

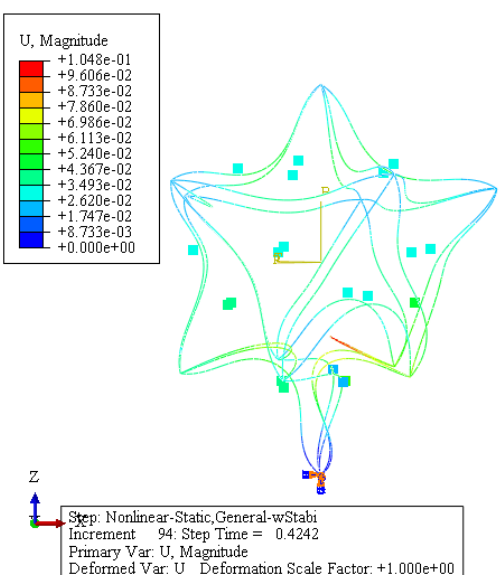


(a) Boundary Condition 1 (b) Boundary Condition 2 (c) Boundary Condition 3

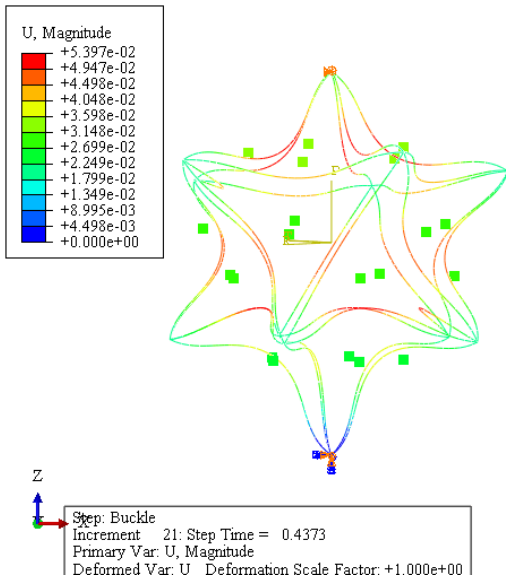
Figure 38: Frame Boundary Conditions.Yellow $x's$ represent reference points, and arrows equivalent loads. Tracked points are used to establish the force versus displacement response.

response, therefore restricting the rotational DOF and the vertical displacement DOF causes unsymmetrical response. Furthermore, the unsymmetrical BC ran into numerical convergence problems after about $\sim 45\%$ of the equivalent SL pressure, a common issue when the Newton Raphson technique runs into a bifurcation point.

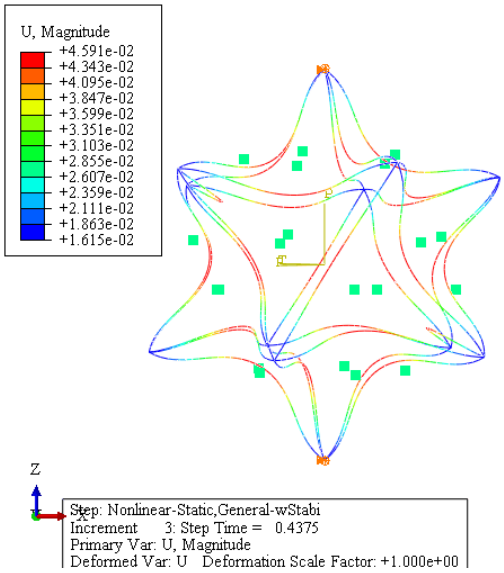
To visualize nonlinear behavior as pressure increases, the latter is plotted against one of the edges' midpoints located adjacent to the bottom vertex, where the unsymmetrical behavior occurs. Results plotted for the three BC are shown in Figure 40. Note in Figure 40a that just before the analysis stops, a snap-back like behavior is shown, followed by a zero slope that drives numerical convergence issues. The snap-back like behavior shown indicates a beam withdrawal or change in displacement direction while still taking load. Even though the slope reverses, there is no softening, therefore the beam does not collapse. For the BC 2, where the frame starts responding slightly more



(a) Boundary Condition 1

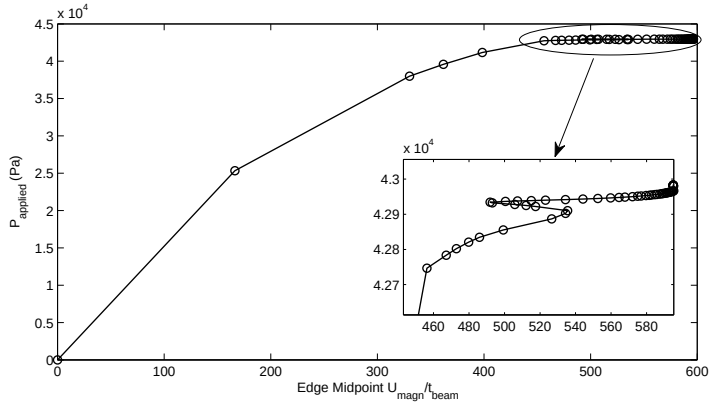


(b) Boundary Condition 2

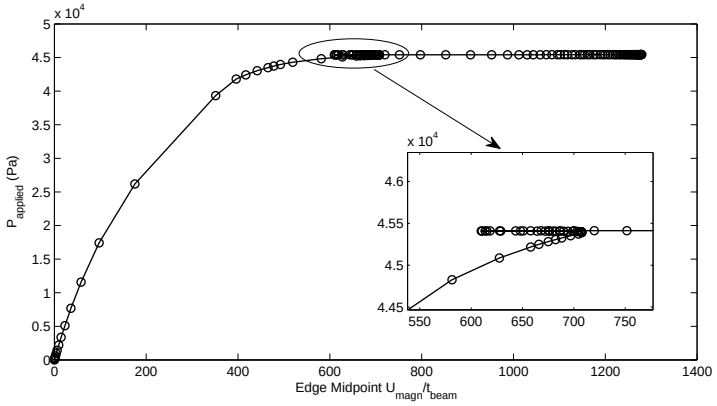


(c) Boundary Condition 3

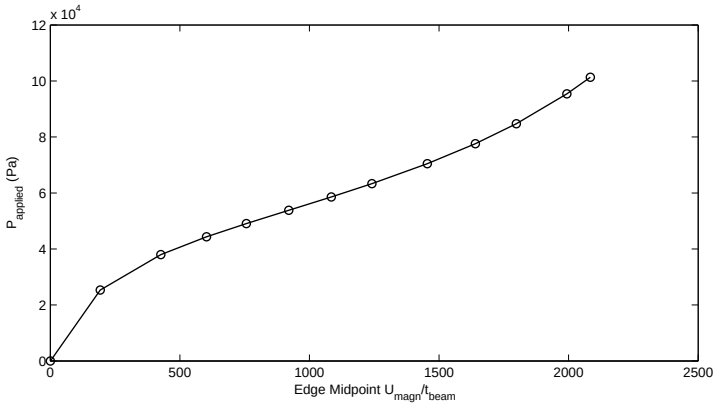
Figure 39: Frame Boundary Conditions Comparison - Displacement Contours at ~ 43% of the equivalent SL pressure.



(a) Boundary Condition 1



(b) Boundary Condition 2



(c) Boundary Condition 3

Figure 40: Frame Boundary Conditions Comparison - Edge Midpoint vs Equivalent Pressure

symmetrically, a higher slope suggests that the frame increases its stiffness as it approaches to a symmetrical configuration (Figure 40b). Nevertheless, snap-back is seen just before approaching a zero slope. Note that in this case, the snap-back is more pronounced, showing a slope closer to zero around that area. The snap-back can be attributed to the fixed BC 'pulling' the beam back in order to achieve an equilibrium state. Not finding the equilibrium path, the beam snaps-forward short after the stiffness matrix becomes singular, running into convergence problems. On the other hand, the frame's response using the BC 3 is not only symmetrical, but also its stiffness increases significantly and the analysis fully converges (Figure 40c). Therefore, BC 3 is selected for further analyses.

3.6.4 Convergence Study.

This convergence study was performed with a buckling analysis, where the first five critical pressures were used to establish convergence. The model was established using the frame standalone with the coupling constraint and the symmetric BC discussed in Section 3.6.3. The edges are seeded homogeneously, from 5 to 25 seeds, corresponding to 150 to 750 seeds. Two methods were used to evaluate convergence. First, the maximum % difference of all critical pressures is considered by comparing the i seeding value against the $i-1$ value. Second, the maximum % difference of all the critical pressures was considered, this time by comparing the i seeding value against the last, i_{end}, value. Tabulated results, included in Section A.7, show that 270 elements corresponding to a seeding of 8 per member is sufficient to achieve convergence within 0.01%.

3.6.5 Beam Profile Study.

The last frame study is the beam profile. This study pretends to answer the last question formulated at the beginning of this chapter, repeated here: considering the frame performance, which is better, hollow or solid beams? At this point, the appropriate number of elements and BC were known. Those were used in this study with a linear static analysis to evaluate the frame's response to the changes in the beam's circular profile, by adjusting the beam's thickness to radius ratio (c). A profile comparison for c values of 0.05, 0.5, and 0.95 is shown in Figure 41. Note that since the mass is held constant, the radius changes inversely proportional to c. Linear analysis provides a rough approximation of the frame's response, but since the interest is to evaluate the response with respect to the solid beam, it provides an efficient way to do so. Therefore, the results considered were normalized to the solid beam, which include moment of inertia (I_n) , maximum stress ($S_{max,n}$), maximum displacement ($U_{max,n}$) and critical pressure ($P_{crit,n}$). The study was performed by changing the beam's thickness to radius ratio, c, as: $0.05 \leq c \leq 0.95$.

The 'perfect' frame has the highest moment of inertia and critical pressure, and the lowest displacement and stress. Each value is plotted against c in Figure 42. Note that the best frame performance is achieved as c tends to 0. In reality, such value is unattainable and considering that the minimum manufacturable thickness is material dependent, a $c = 0.05$ is selected for the icosahedron analysis. The tabulated results of this study are included in Section A.8.

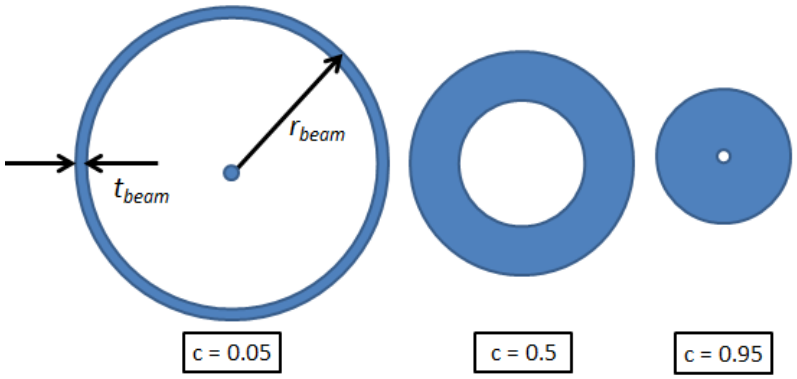


Figure 41: Beam Profile Comparison (not to scale)

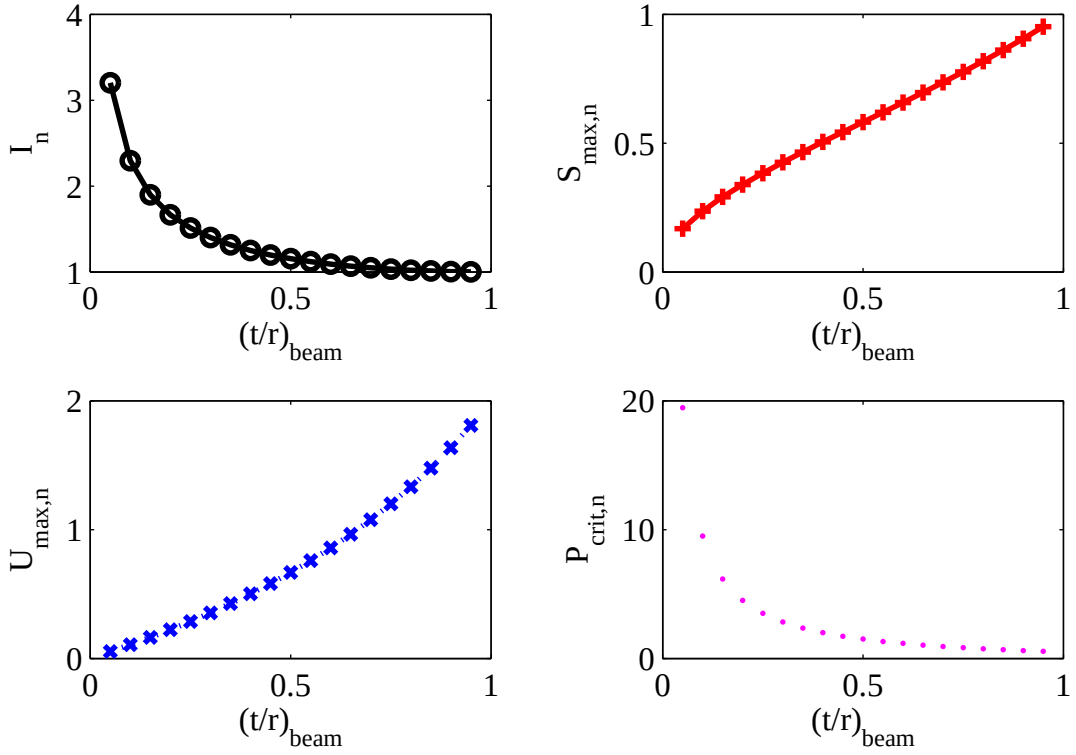


Figure 42: Beam Profile Study. I_n, $S_{max,n}$, $U_{max,n}$ and $P_{crit,n}$ refer to the moment of inertia, maximum stress, maximum displacement and critical pressure, all normalized by the solid beam results. t and r refer to beam's thickness and radius.

3.7 Icosahedron

At this point, several studies have been conducted in order to validate the selected FE techniques, as well as to find the correct modeling discretization and make design decisions. These studies provided the background needed to properly model the icosahedral skin and frame combination; results were gathered in order to establish the basic model. Nonetheless, there are several techniques that are particular for the icosahedron. Models analyzed are presented following the discussion of such techniques.

3.7.1 Modeling Techniques.

Additional to the techniques verified during previous studies, three more techniques are used for the icosahedron. First, the skin connectivity to the frame. Abaqus provides various methods to model such connectivities, including the contact algorithm and the tie constraint. The latter, used in the icosahedron model, ties two surfaces together during analysis. The tie constraint allows for the selection of specific DOF to be tied, and is capable of tying beam elements to surface elements, such as the shell and membrane elements previously discussed. The tie is based on master and slave surfaces selected by the user; once the DOF to be tied are selected, the constraint eliminates those from the slave surfaces. In the icosahedron case, the frame is the master surface and skin edges are the slave surface, which are coincident to the frame. The constraint accounts for the thickness of the shell or membrane, but ignores the beam's profile. Since membrane behavior is being modeled for the skin, only the displacement DOF are tied.

Second, the complexity of the icosahedron introduced convergence issues that were resolved by editing FE processor defaults. Referring back to the Newton Raphson discussion presented in Section 2.4.2, the FE processor starts with an initial load

increment, which is a percentage of the total load applied to the structure. That initial increment can be provided by the user, or can be left as the default; nonetheless is arbitrary. Depending on the nonlinearity of the problem, that increment might not be appropriate to attain equilibrium, therefore an iterative process is needed in order to adjust the increment such that equilibrium can be achieved. Abaqus has an algorithm that controls the iterative process to aid convergence, but allows users to control most parameters. By default, the algorithm changes the initial load increment by 25 to 75% every time for up to five iterations. The same iterative process can occur for any load increment, as required. If an equilibrium solution is not found by the fifth iteration, the processor stops. Additionally, equilibrium equations are calculated several times within each iteration as part of the Newton Rahpson process.

During the analysis of the icosahedron, three parameters related to the analysis algorithm were changed: the amount of maximum iterations per load increment, the % increment change per iteration, and the amount of equilibrium calculations made before moving to another iteration. Due to the model sensitivity to the given initial increment, the amount of maximum iterations per load increment was adjusted from 5 to 25, allowing the processor to change the initial increment more times without stopping. Furthermore, the increment change per iteration was changed to no less that 50%. Additionally, the maximum number of equilibrium calculations was doubled. All these changes made the analysis less sensitive to both initial increment and rapid changes in slope, thus enabling a solution path.

Third, the unsymmetric matrix storage was used. Abaqus provides the option of storing the entire tangent stiffness matrix or just its symmetric part during the analysis.

For most static uncoupled analyses, storing the symmetric part provides an efficient mean of analysis. But there are certain analyses that make the matrix unsymmetric, therefore assuming that is symmetric can make the analysis run into convergence issues. One of the cases that produces an unsymmetric matrix is the use of follower loads in large displacement analysis [12, Ch. 32.4.2].In the icosahedron case, large displacement is caused by applying pressure to the skin (a follower load). Therefore, storing the entire matrix indeed aided solution convergence, even though it was not needed when considering simple membranes.

These techniques, along with results from previous studies, were gathered to develop a basic icosahedron model. Models with different material properties were then developed based on the basic model.

3.7.2 Models.

The basic model is a conglomerate of previously stated techniques and design features. The following modeling techniques and properties are shared in all icosahedral models considered for analysis:

- Dimensionality: A fixed diameter of 0.3048 m (1 ft.) was selected, with a beam thickness to radius ratio, c, equal to 0.05. The beam radii and skin thicknesses were derived for a desired W/B, using Equation (2.19) and Equation (2.15), respectively.
- Load: the load was set at SL pressure (101,325 Pa).
- Boundary Conditions: the BC that produced symmetry was selected (BC 3 in Figure 38c). The symmetric BC was composed by fixing the displacement DOF plane: U1=U2=0, of opposite vertices.

- Mesh: the mesh was composed of M3D3 membrane elements for the skin and B32 beam elements for the frame.

- Analysis: the Newton Raphson technique with adaptive automatic stabilization was selected along with unsymmetric matrix storage. Also, solution controls were adjusted to aid convergence. A linear buckling analysis was conducted in one of the models in order to visualize buckling mode shapes.

- Constraint: the tie constraint was used to connect the skin to the frame by only tying the displacement DOF.

A representation of the basic model is shown in Figure 43, where arrows represent the pressure applied to the skin and orange symbols at the top and bottom vertices represent the symmetric BC (BC3 in Section 3.6.3). Note how the skin is tied at the mid-plane of beams, such that half of the beam cross-sections is exposed. Using the basic model, seven models were developed by changing the material properties and desired W/B. Material properties related to three materials were selected, as shown in Table 1: # 5, # 6 and # 10. Material #5 was selected because, though still in research, provides the best combination of specific stiffness and strength. On the other hand, material # 6 is the weakest of the three selected, but is a material well researched, with linear behavior and commercially available. Material # 10 provides a middle ground between the other two in terms of strength and stiffness. Materials and buoyancy (acW/B) of models developed are shown in Table 5.

Note in Table 5 that the first five models have a desired W/B of 0.9, as the first three have the material properties of selected materials for both skin and frame. Models 4 and 5

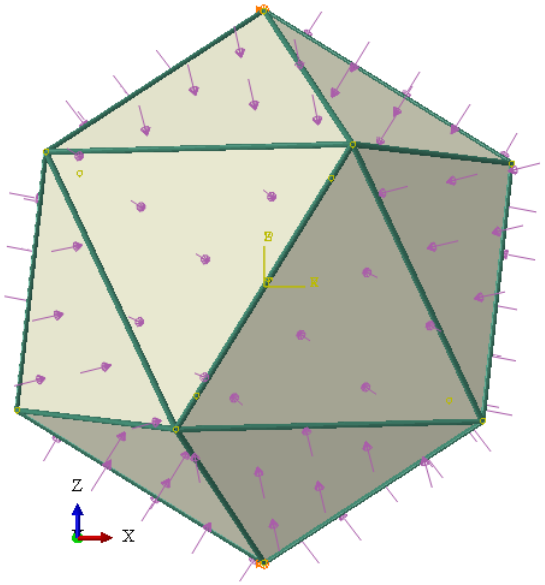


Figure 43: Icosahedron Basic Model. Arrows represent the pressure applied to the skin and orange symbols at top and bottom vertices represent the symmetric BC.

Table 5: Icosahedron Models

	Material		**Desired W/B**				
Model	***Frame***	***Skin***	***Frame***	***Skin***	r_{beam}	t_{skin}	***# Ele.***
1	6 (Beryllium)	6 (Beryllium)	0.5	0.4	1.41E-03	1.05E-05	7020
2	10 (Spectra)	10 (Spectra)	0.5	0.4	1.95E-03	2.00E-05	8600
3	5 (CNT)	5 (CNT)	0.5	0.4	1.49E-03	1.18E-05	7020
4	6 (Beryllium)	5 (CNT)	0.5	0.4	1.41E-03	1.18E-05	7020
5	10 (Spectra)	5 (CNT)	0.5	0.4	1.95E-03	1.18E-05	8600
6	10 (Spectra)	10 (Spectra)	0.4	0.4	1.74E-03	2.00E-05	7020
7	5 (CNT)	5 (CNT)	0.4	0.4	1.33E-03	1.18E-05	8600

are composed of hybrid combinations of Beryllium and Spectra fiber frames with CNT skin, respectively; thus providing stiffened versions of models 1 and 2, respectively. The

last two models have a desired W/B of 0.8, composed of Spectra fiber and CNT material properties, respectively.

The analysis process used to evaluate icosahedron models is slightly different from the one discussed in Section 3.2. The detailed process is shown in Figure 44, along with the name of the Matlab functions and Python codes used and their respective locations in the appendices.

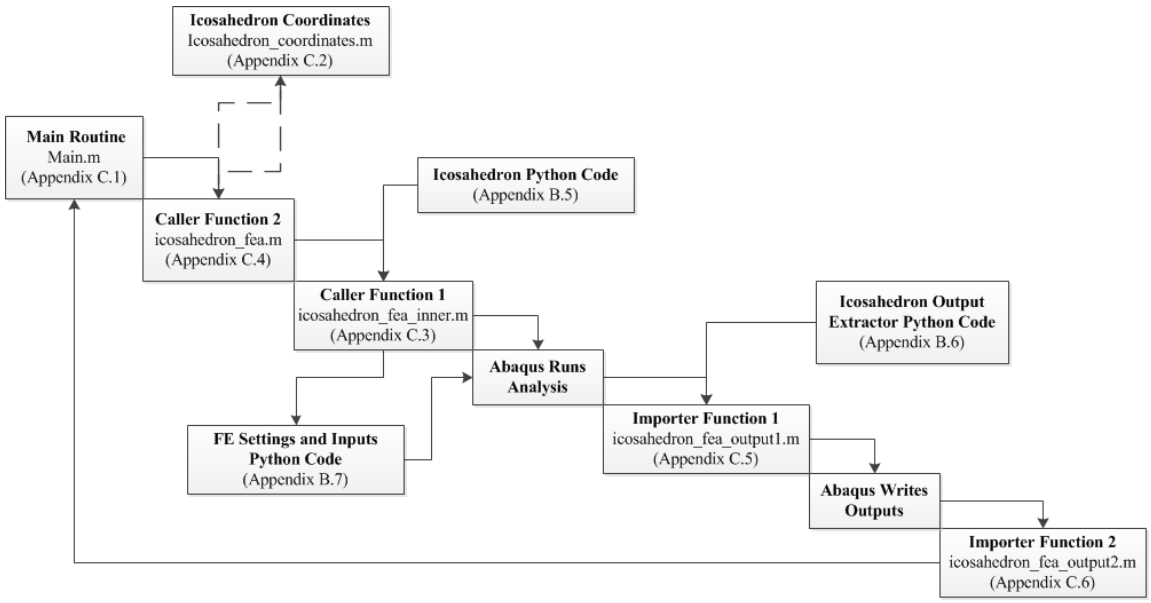


Figure 44: Icosahedron Analysis Process Diagram

Note that the Python codes with the model and modeling details are established prior to the analysis process. Matlab's caller 1 and importer 1 functions send the respective codes to Abaqus for analysis and output extraction. The importer function 2 takes the outputs extracted from Abaqus and imports them into Matlab as a structure. All Python codes and Matlab functions are included in Appendix B and Appendix C, respectively.

3.8 Summary

The purpose of this chapter is to discuss different studies that not only validated the proposed FE techniques, but also enabled the understanding of some icosahedron features and supported some design and modeling decisions. Convergence studies of square and circular membranes, validated the Newton Raphson analysis technique, demonstrated the usefulness of the stabilization mechanism in initially flat and stress free membranes, and validated the membrane (M3D3) element. The thickness study of the circular membrane demonstrated the capacity of the shell (S3R) element to behave as a membrane for small thicknesses. These studies established leeway for skin equivalent triangle studies. The material properties study was particularly important because it defined the relationship between skin behavior and changes in material properties. Furthermore, it addressed an important question: the effect of material properties in W/B. It demonstrated that the most relevant material properties are modulus of elasticity and density, in that order; and that both have minimal effect on the W/B ratio after deformation.

The frame studies identified the proper boundary condition to achieve symmetry during modeling analysis, as well as the improved performance of the hollow beam over the solid. Additionally, the coupling constraint and its use were established and studied. Finally, all study results were gathered along with techniques that aided convergence, and used for modeling the icosahedron. Seven icosahedron models were developed for analysis by considering a combination of material properties and W/B ratios.

IV. Results and Discussion

4.1 Overview

Modeling development techniques and their validation were discussed in Chapter 3, along with the results of the different studies leading to the icosahedron model. In this chapter, those modeling techniques are used to: (1) evaluate the conservation of symmetry for the nonlinear analysis of the icosahedron, (2) evaluate the linear buckling response of the icosahedron, and (3) compare the nonlinear response of the icosahedron for the seven models described in Table 5. The latter includes a convergence history case study, the buoyancy effects and the structural response of the models.

4.2 Symmetry Validation

The validation of symmetry is an important analysis tool because: (1) a symmetrical distribution of loads and stress is achieved, improving structural response, and (2) it allows for the use of critical design points that are independent on a specific structure's face or beam to represent its response, therefore reducing the amount of representative data needed. Therefore, before considering the response of icosahedral models, the symmetry is evaluated by comparing the coordinates of each vertex and face center before and after deformation. Each vertex and face center is represented by a node in the discretized model (see Figure 45a). Node displacements are characterized by the difference between their initial and final coordinates. In the case of spherical coordinates (see Figure 34), symmetry occurs when angles, θ and ϕ, remain constant before and after deformation, allowing deformation only in the radial (r) direction. Furthermore, vertices

would share the same radial displacement; and equally with face centers. Model 3 was taken as an example to evaluate symmetry (see Table 5). Initial and final spherical coordinates of each vertex are shown in Table 6, along with the % difference in both angles. Note that the final radial coordinate is the same within four decimal places for all vertices and angular symmetry is conserved within 0.03%. Furthermore, coordinates of each face center are compared in Table 7. In this case, the skin deviates slightly from symmetry, but the final radial coordinate is the same within four decimal places, too, and angular symmetry is conserved within 0.1%.

Table 6: Vertices Symmetry in Model 3

Vertex	**Initial**			**Final**			$\boldsymbol{\theta}$ **%D**	$\boldsymbol{\phi}$ **%D**
	$\theta\,(deg)$	$\phi\,(deg)$	$r\,(m)$	$\theta\,(deg)$	$\phi\,(deg)$	$r\,(m)$		
1	0.00	90.00	0.1524	0.000	90.000	0.1519	0.000%	0.000%
2	0.00	26.57	0.1524	0.000	26.556	0.1519	0.000%	0.033%
3	72.00	26.57	0.1524	72.001	26.556	0.1519	0.001%	0.034%
4	144.00	26.57	0.1524	144.001	26.556	0.1519	0.001%	0.034%
5	-144.00	26.57	0.1524	-144.001	26.556	0.1519	0.001%	0.034%
6	-72.00	26.57	0.1524	-72.001	26.556	0.1519	0.001%	0.034%
7	36.00	-26.57	0.1524	35.999	-26.570	0.1519	0.003%	0.019%
8	108.00	-26.57	0.1524	107.999	-26.570	0.1519	0.001%	0.020%
9	180.00	-26.57	0.1524	180.000	-26.570	0.1519	0.000%	0.020%
10	-108.00	-26.57	0.1524	-107.999	-26.570	0.1519	0.001%	0.020%
11	-36.00	-26.57	0.1524	-35.999	-26.570	0.1519	0.003%	0.019%
12	0.00	-90.00	0.1524	143.999	-90.000	0.1519	0.000%	0.000%

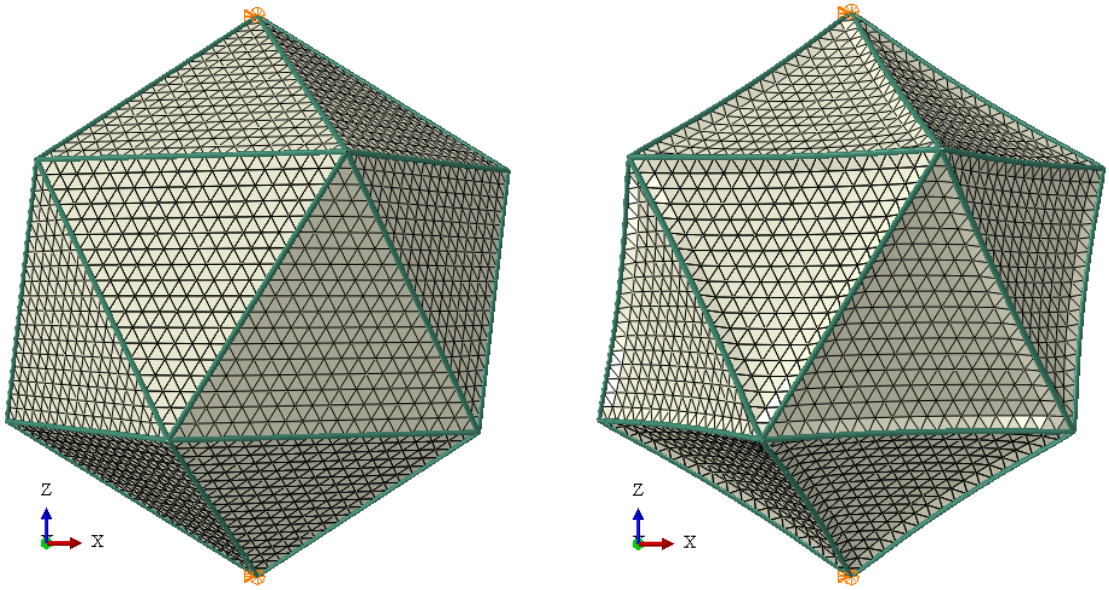
$\%\boldsymbol{D}$ refers to the % difference of either θ or ϕ.

Table 7: Face Center Symmetry in Model 3

Face Center	**Initial**			**Final**			$\boldsymbol{\theta}$ **%D**	$\boldsymbol{\phi}$ **%D**
	$\theta\,(deg)$	$\phi\,(deg)$	$r\,(m)$	$\theta\,(deg)$	$\phi\,(deg)$	$r\,(m)$		
1	36.00	52.6	0.1211	35.997	52.617	0.1125	0.009%	0.011%
2	108.00	52.6	0.1211	108.001	52.617	0.1125	0.001%	0.011%
3	180.00	52.6	0.1211	180.000	52.616	0.1125	0.000%	0.012%
4	-108.00	52.6	0.1211	-108.001	52.617	0.1125	0.001%	0.011%
5	-36.00	52.6	0.1211	-35.997	52.617	0.1125	0.009%	0.011%
6	72.00	-52.6	0.1211	71.999	-52.630	0.1125	0.002%	0.014%
7	144.00	-52.6	0.1211	144.003	-52.630	0.1125	0.002%	0.014%
8	-144.00	-52.6	0.1211	-144.003	-52.630	0.1125	0.002%	0.014%
9	-72.00	-52.6	0.1211	-71.999	-52.630	0.1125	0.002%	0.014%
10	0.00	-52.6	0.1211	0.000	-52.629	0.1125	0.000%	0.013%
11	36.00	10.8	0.1211	35.999	10.802	0.1125	0.003%	0.092%
12	72.00	-10.8	0.1211	71.999	-10.824	0.1125	0.001%	0.107%
13	108.00	10.8	0.1211	108.001	10.803	0.1125	0.001%	0.088%
14	144.00	-10.8	0.1211	144.001	-10.823	0.1125	0.001%	0.102%
15	180.00	10.8	0.1211	180.000	10.802	0.1125	0.000%	0.094%
16	-144.00	-10.8	0.1211	-144.001	-10.823	0.1125	0.001%	0.102%
17	-108.00	10.8	0.1211	-108.001	10.803	0.1125	0.001%	0.088%
18	-72.00	-10.8	0.1211	-71.999	-10.824	0.1125	0.001%	0.107%
19	-36.00	10.8	0.1211	-35.999	10.802	0.1125	0.003%	0.092%
20	0.00	-10.8	0.1211	0.000	-10.823	0.1125	0.000%	0.100%

$\%\boldsymbol{D}$ refers to the % difference of either θ or ϕ.

Initial and final states of Model 3 are shown in Figure 45a and Figure 45b, respectively. As previously mentioned, the skin is tied to the beams axes, which are located at the center of each beam. The tie constraint accounts for the skin thickness, even though it is not captured by the contours. Now that symmetry has been verified both visually and numerically, the nonlinear response of all models can be compared. But first, let's consider the mode shapes and critical pressure predicted by the linear buckling analysis.



(a) Initial State (b) Final State

Figure 45: Icosahedron: Model 3 States

4.3 Icosahedron Linear Buckling Analysis

A linear buckling analysis was conducted primarily to visualize the possible buckling modes shapes. In this case, the S3R shell element was used do to its bending terms that the membrane element lacks. Previous studies indicated that the shell element

behaves like a membrane for skin thicknesses below 0.2 mm, but regardless of that behavior, the shell still carries bending terms, numerically speaking. The linear buckling analysis provided by Abaqus does not have the automatic stabilization that the static analysis does, thus not allowing for the initially flat membrane to acquire stiffness. This results in numerical singularities. Bending terms carried by the shell element allows the buckling analysis to estimate mode shapes and buckling loads without running into such singularities. The issue behind using such analysis is that membrane stiffness will not be captured and, being membrane stiffness the primary driver of the skin behavior, modes shapes are expected to be underestimated.

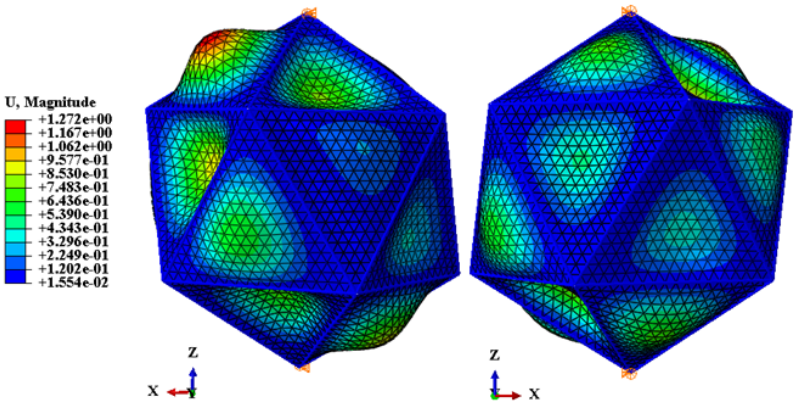
In theory, one can calculate as many buckling modes as DOF in the FE model, but most often than not, only the first buckling mode is relevant since higher modes have no chance of taking place before the structure collapses [27]. Nonetheless, in order to see if the analysis predicts frame buckling, several buckling modes were considered. Buckling modes 1, 2 and 14 are shown in Figure 48 for Model 3. Is it clear that all the predicted modes are skin related. Note that displacement values are relative to the initial configuration, therefore they do not represent actual displacements. As the critical load increases, more complex mode shapes appear, as shown in Figure 46c, but the frame does not seem to be affected by them. Furthermore, there was no frame buckling predicted within the first 200 modes. Having so many skin related modes can be associated to the numerical bending carried by the shell element, such that even though it is insignificant when considering a nonlinear static analysis, it becomes the driver of such modes. Therefore, predicted critical loads are considerably low compared to the SL pressure (101,325 Pa).

What is shown in Figure 48 can be interpreted as a series of in and out of plane deflections on the skin caused by compressive forces developed through bending, where triangle interconnections represent a series of folded plates. The latter cause opposite deflections in adjacent planes, such that symmetry is conserved. Since the frame is considerably stiff, compared to the skin bending stiffness provided by the shell element, no frame buckling modes are detected. Therefore, this analysis proves not to be representative of the icosahedral skin response.

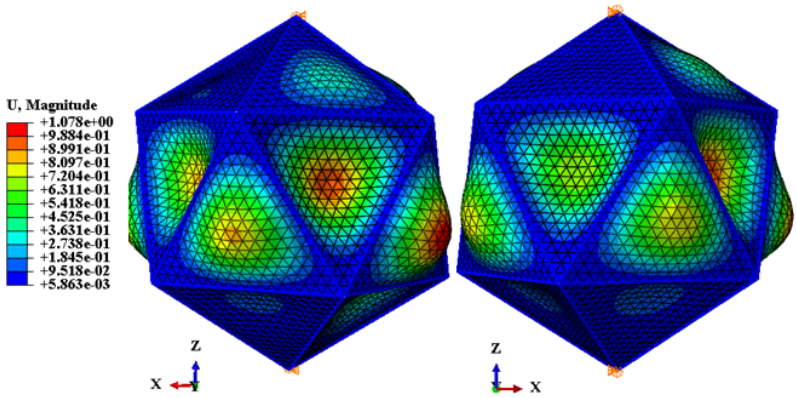
The following section considers the nonlinear static response of the different models, not only allowing for membrane to acquire stiffness, but establishing the effect of the frame on the skin and the overall stiffness of the models.

4.4 Icosahedron Nonlinear Static Analysis

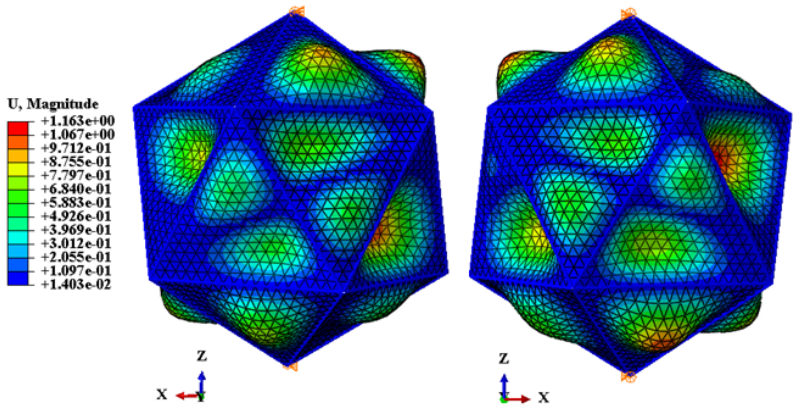
The structural analysis of the icosahedron not only provides insight on the structural response, but also the means to evaluate the effects of such response in its buoyancy. Therefore, two main aspects are considered here: structure's response and its buoyancy. The structural response is characterized by displacements and stresses exhibited by the structure as incremental pressure is being applied. In a static fashion (rather than dynamic), the Newton Raphson technique enables the capture of such response as incremental pressure is being applied to the skin, including nonlinear effects. But internal volume is lost as a byproduct of the structures deflection, affecting its buoyancy. The latter is particularly important for the LTAV application. Before considering such aspects, the case study of the Newton Raphson convergence history is presented in order to establish the iterative process that was needed to find the solution path.



(a) Mode 1: $P_{crit} = 6.98\ Pa$



(b) Mode 2: $P_{crit} = 7.26\ Pa$



(c) Mode 14: $P_{crit} = 20.5\ Pa$

Figure 46: Buckling Modes in Model 3

4.4.1 Convergence History.

The convergence history of a nonlinear problem is unique and dependent on the solution path that establishes equilibrium. As discussed in Section 2.4.2 and Section 3.7.1, the nonlinear static analysis implemented in Abaqus uses a series of load (also called 'time') increments to find the solution path. Since the solution path is unknown, the Newton Raphson technique is used in order to find each point within the solution path that satisfies equilibrium. Given the equilibrium at time increment $t - 1$, the time increment t is selected and a number of iterations follow to try and find the next equilibrium state, referred as *equilibrium iterations*. If the maximum number of equilibrium iterations is reached before achieving the equilibrium state, the time increment is reduced and the equilibrium iterations start again. This process is repeated up to the point that equilibrium is found or the maximum *number of attempts* is reached.

The icosahedron models followed this process until the equilibrium path was found for a SL pressure. The convergence history of model 3 (see Table 5 for the model's description) is shown in Figure 47 as a case study. First, the number (#) of attempts made prior to finding equilibrium per increment is presented. It is followed by the number of equilibrium iterations per increment (middle). Finally, the load increment for each successful equilibrium state per increment is displayed. Note in the first increment that 19 attempts were made prior to finding equilibrium, indicating that the initially guessed time increment was far off from the increment needed for equilibrium. Once equilibrium is found in the first increment, the amount of attempts reduces to 15 for the second, still a high number, resulting in even more reduction of time increment from $\sim 10^{-5}$ to $\sim 10^{-10}$. After this point, an steady increase in time increment is clear and the number of attempts

reduces to 1, with the exception of the next to last increment. On the other hand, the number of equilibrium iterations, though dependent on time increments, is associated more with the iterative technique used in the Newton Raphson to find the solution point, though seeing a different variation (mid-plot).

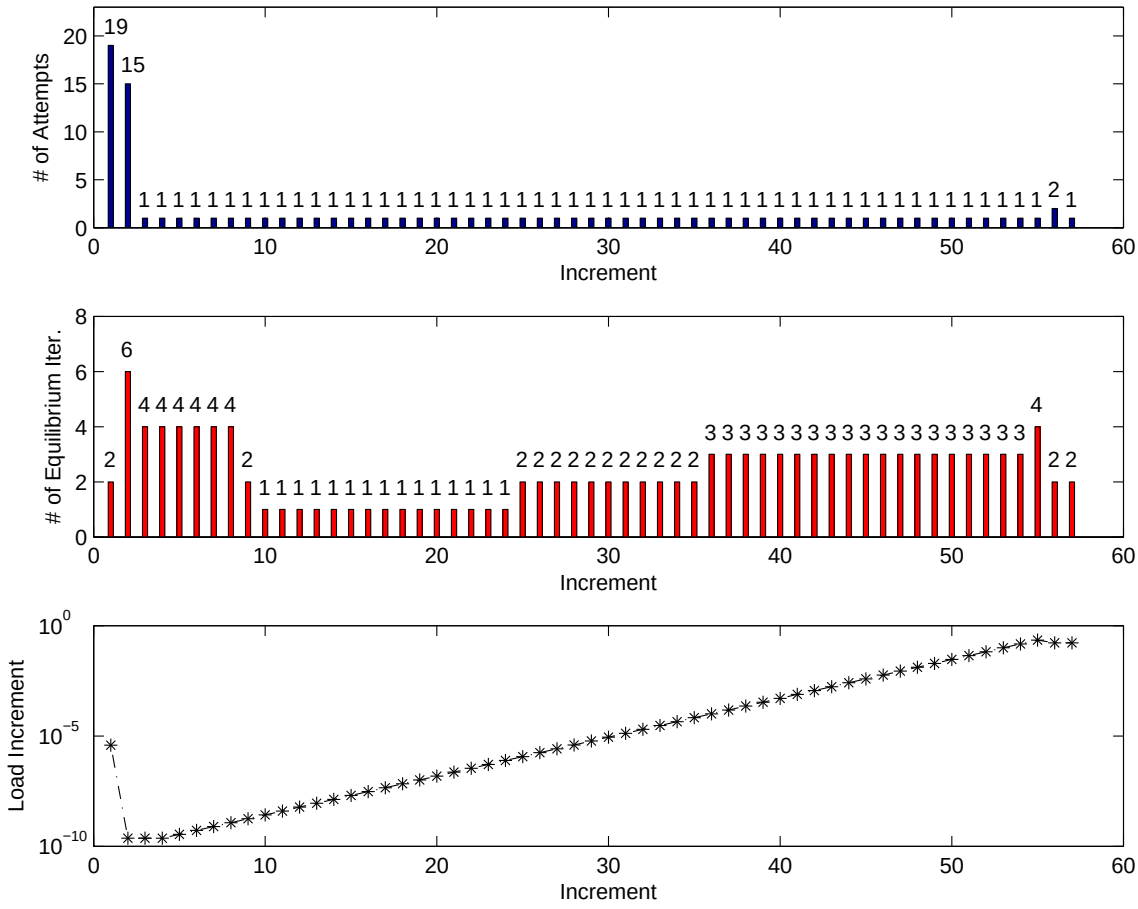


Figure 47: Convergence History of Model 3. Number (#) of attempts to per increment (top). # of equilibrium iterations per increment (middle). Load increment per increment (bottom).

This convergence history was common within all seven icosahedron models. The models' sensitivity to low time increments within the first portion of the history is related to the nonlinearity that the membrane brings as it is acquiring stiffness. This behavior will be discussed in the structural response section (Section 4.4.3).

4.4.2 Buoyancy Effects.

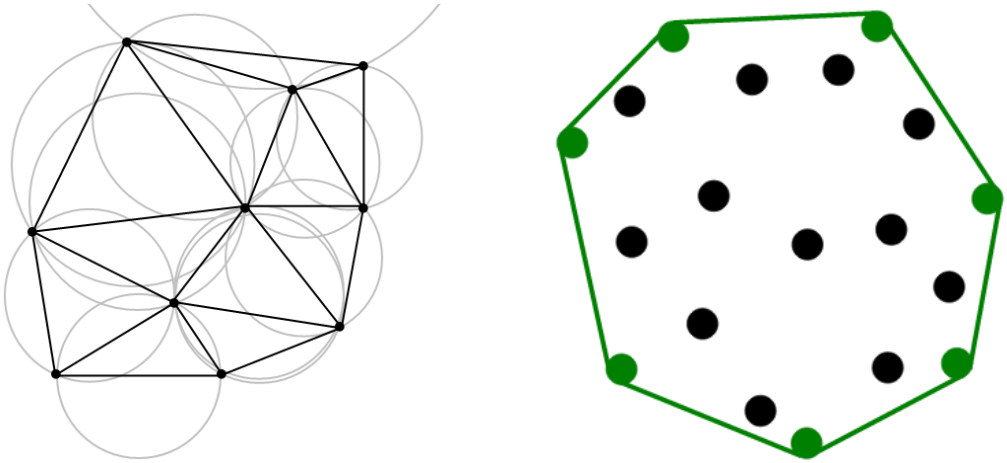
Two aspects are important when considering buoyancy effects: volume reduction and applied pressure. The volume reduction is in the denominator of the W/B equation (see Equation (2.21)), restated below:

$$\frac{W}{B} = \frac{9.5745 t_{skin}\, r^2\, \rho_{skin} + 99.098\left(2c - c^2\right) r_{beam}^2\, r\, \rho_{frame}}{\left[2.5362 r^3 - V_r\right]\left(\frac{P_{air,o}}{RT_{air,o}}\right)} + \frac{P_{air,i}}{P_{air,o}}\frac{T_{air,o}}{T_{air,i}} \qquad \text{(2.21 revisited)}$$

Therefore, it contributes negatively to the buoyancy. The volume reduction is given by the initial volume minus the volume at each load increment. Once deflected, the volume of the structure is calculated using Matlab functions: convhull.m [32] and delaunayTriangulation.m [33]. This is done by inputing nodal displacements of each increment into the *delaunayTriangulation* function, creating a triangulated surface. In two dimensions, triangulation is the division of a surface into a set of triangles with each side shaded with two adjacent triangles as a restriction [55]. A delaunay triangulation is then a triangulation of a set of points such that no point in the set lies inside the circumcircle of any of the triangles [19]. This concept is shown in Figure 48a. The delaunay triangulation is then inputed into the *convhull* function, solving for the internal volume at each increment.

Given a set of points, a convex hull is defined as the minimum convex subset that contains that set. For example, a set of points in Figure 48b. The minimum set that

encapsulate all points are the set of green points or the convex hull. These two concepts can be applied three-dimensionally. In the icosahedron case, the points are the element's nodes, and the convex hull is indeed the icosahedron. Knowing node locations at each increment enables the internal volume calculation as the skin deflects.



(a) Delaunay Triangulation [24] (b) Convex Hull [25]

Figure 48: Representation of the Volume Calculation Techniques

In order to calculate the volume reduction, the initial volume is subtracted from the volume at each increment. The initial volume obtained with these functions was verified against the analytical equation (Equation (2.12)), and both yielded the same result.

Applied pressure versus volume reduction normalized by the initial volume is shown in Figure 49 for all seven models (model descriptions are presented here, again for convenience, in Table 5). The horizontal dashed line represents a feasible vacuum, since a perfect vacuum is not possible to achieve, but one -commonly referred to as an ultra high

Table 5: Icosahedron Models

	Material		Desired W/B				
Model	***Frame***	***Skin***	***Frame***	***Skin***	r_{beam}	t_{skin}	***# Ele.***
1	6 (Beryllium)	6 (Beryllium)	0.5	0.4	1.41E-03	1.05E-05	7020
2	10 (Spectra)	10 (Spectra)	0.5	0.4	1.95E-03	2.00E-05	8600
3	5 (CNT)	5 (CNT)	0.5	0.4	1.49E-03	1.18E-05	7020
4	6 (Beryllium)	5 (CNT)	0.5	0.4	1.41E-03	1.18E-05	7020
5	10 (Spectra)	5 (CNT)	0.5	0.4	1.95E-03	1.18E-05	8600
6	10 (Spectra)	10 (Spectra)	0.4	0.4	1.74E-03	2.00E-05	7020
7	5 (CNT)	5 (CNT)	0.4	0.4	1.33E-03	1.18E-05	8600

vacuum- can be obtained within 1e - 7 Pa of it [40]. Volume reductions vary no more that 4% between all models, confirming the statement made at the end of Section 3.5.3: that material properties have minimal effect in volume reduction. Nonetheless, it can be seen that stiffer models, models 3 and 7, only suffer $\sim 1.5\%$ volume reduction. Also, those stiffer models show a close to linear relationship with significantly higher slopes, an indicator of their rigidity compared to the rest of the models. The volume reduction can be used as a measure of collapse, and even though this is clearly not the case, using volume reduction provides a globalized method to measure such failure. At this point, the following question arises: what is the effect of volume reduction in structure's buoyancy?

The effect of structural deflection on the W/B depends on two factors: volume reduction and applied pressure. Applied pressure versus W/B are plotted in Figure 50, with the full range of pressures at the bottom and the range pressures that provide a $W/B \leq 1$ at the top. In regards to the effect on volume reduction at the feasible vacuum point, note that models exhibit a W/B equal to the desired W/B (0.9 for Models 1-5, 0.8

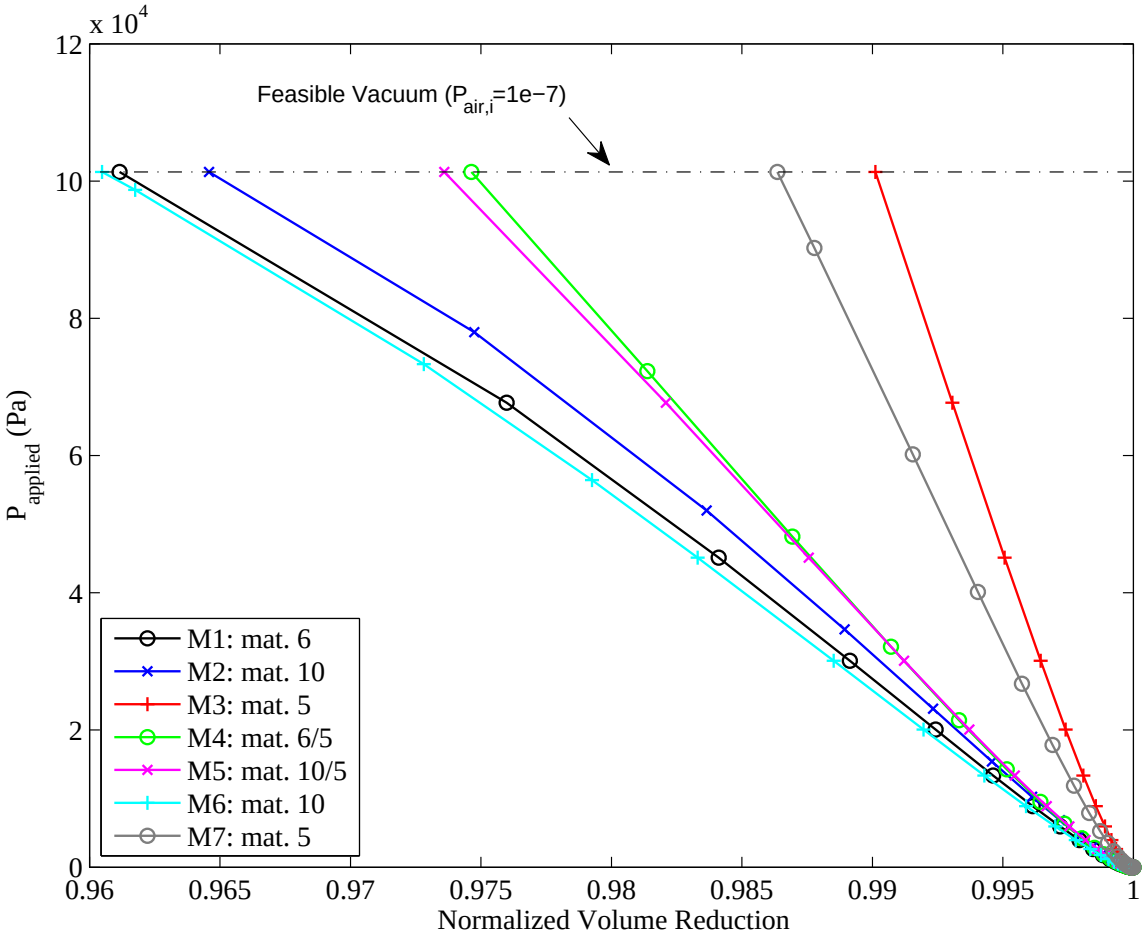


Figure 49: Icosahedron: Applied Pressure versus Normalized Volume Reduction

for Models 6-7; see Table 5) minus the volume reduction for each model. This can also be observed mathematically in Equation (2.21). Note that relatively linear behavior is seen in W/B curves, as a result of the linear behavior of volume reduction. From the point of view of applied pressure, note that all models are buoyant at the range of pressures shown in Figure 50. This bring up an important concept: in order for a vacuum LTAV to be buoyant, a full vacuum is not necessarily required, though desired. The effect of a partial vacuum is reflected in the pressure ratio of the second term in Equation (2.21). As

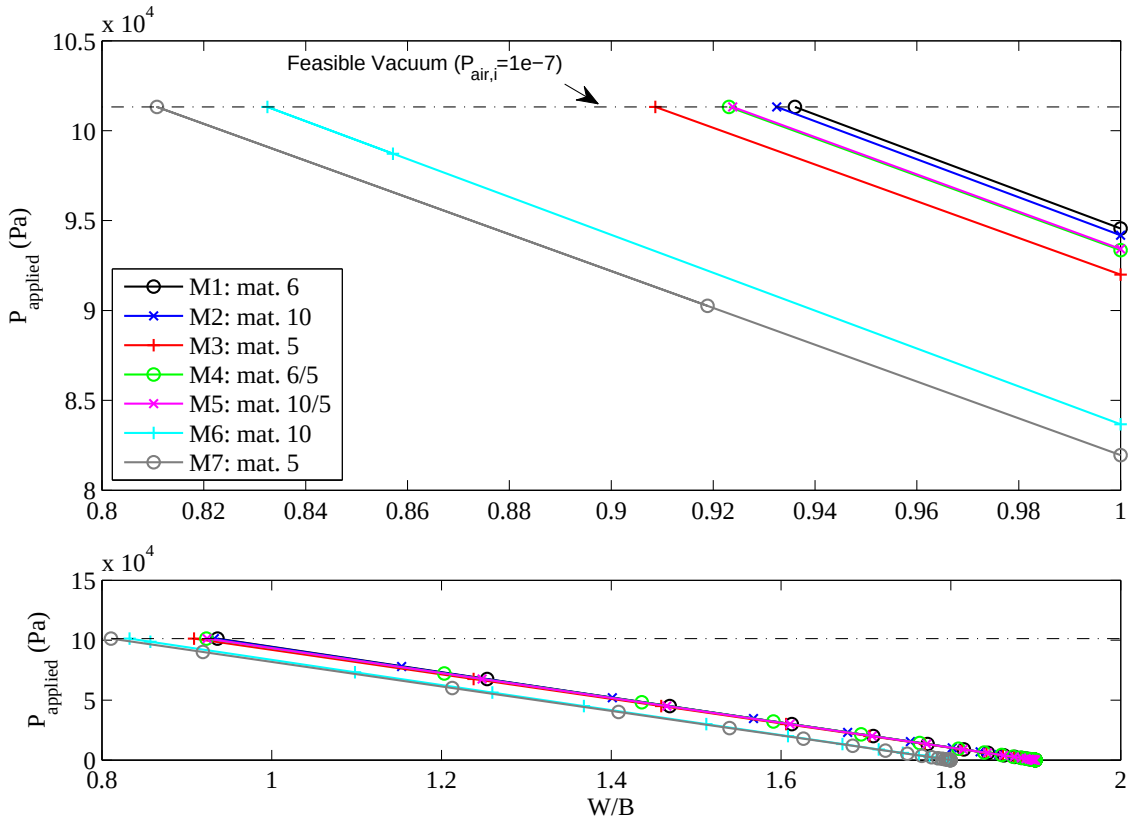


Figure 50: Icosahedron: Applied Pressure versus Weight-to-Buoyancy Ratio. Full analysis (bottom). Close up of full analysis for $W/B \leq 1$.

the internal vacuum is created, the internal pressure, $P_{air,i}$, is reduced, driving that term towards 0, thus increasing the structure's buoyancy. Note that models 1 and 7 achieve neutral buoyancy at 95 kPa and 82 kPa, respectively, thus providing a significant pressure range that can be used to manage the structural load, possible payload added to the vehicle, etc.

Other buoyancy considerations include the change in atmospheric pressure and temperature brought by changes in altitude. This is particularly important to determine the maximum altitude that the structure is capable to achieve before losing buoyancy

(before $W/B = 1$). Comparing models from a W/B point of view, models 6 and 7 have a higher buoyancy as a result of their initially selected W/B of 0.8, instead of the 0.9 value that was initially selected for the first five models. However, volume reduction and W/B figures say very little about the structural behavior of the different models, a consideration that establishes the structure stiffness and possible failure.

4.4.3 Structural Response.

Once buoyancy effects are established and symmetry is verified for all models, only critical points on the design are considered to represent the structural response of icosahedron models. These critical points are, displacement wise: a vertex, edge midpoint and triangle's center. Since both frame and skin share nodes along edges, a vertex and a midpoint represent the behavior of both parts along the edges. First,the applied pressure versus vertex displacement is plotted in Figure 51 and considered for all seven models. The displacement is normalized by the beam diameter of each model. The dashed horizontal line is the feasible vacuum line. The horizontal colored lines represent the points at which each model achieves neutral buoyancy ($W/B = 1$). Note, for example, that models 6 and 7 achieve neutral buoyancy before the first five models. That is because the desired W/B of these models were 0.8, instead of 0.9; and as seen in Figure 50, the W/B does not change considerably. Regardless, color lines represent the 'true' W/B (including the volume reduction). Note that a closely linear relationship is observed for the vertices of all seven models, with deflections in the order of 0.15 to 0.7 times their respective beam diameters. As expected, these displacements are not as pronounced in stiffer models (3 and 7). The slope difference between models 1 and 3 is as much as 114%, a definite proof of how much the material stiffness contributes to the overall

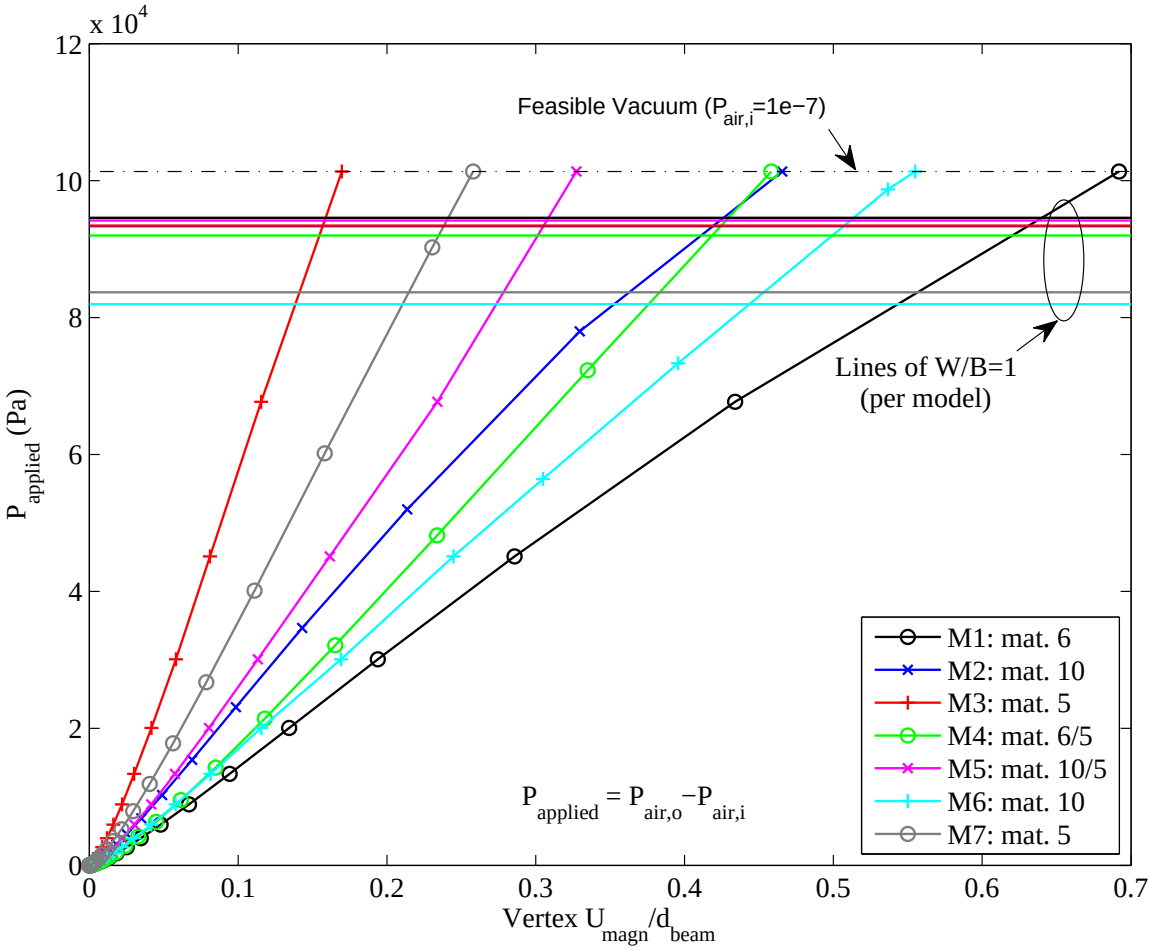


Figure 51: Icosahedron: Applied Pressure versus Vertex Displacement Normalized by the Beam's Diameter

behavior of the structure. Regardless, all vertex displacements show to be less that their diameter, which are not so small to consider them within the linear theory regime, but small enough to validate frame's rigidity.

In a similar fashion, the applied pressure is graphed against an edge midpoint in Figure 52 for all seven models (see Table 5 for model descriptions). Displacements are also normalized by the beam diameter of each model. As expected, larger displacements

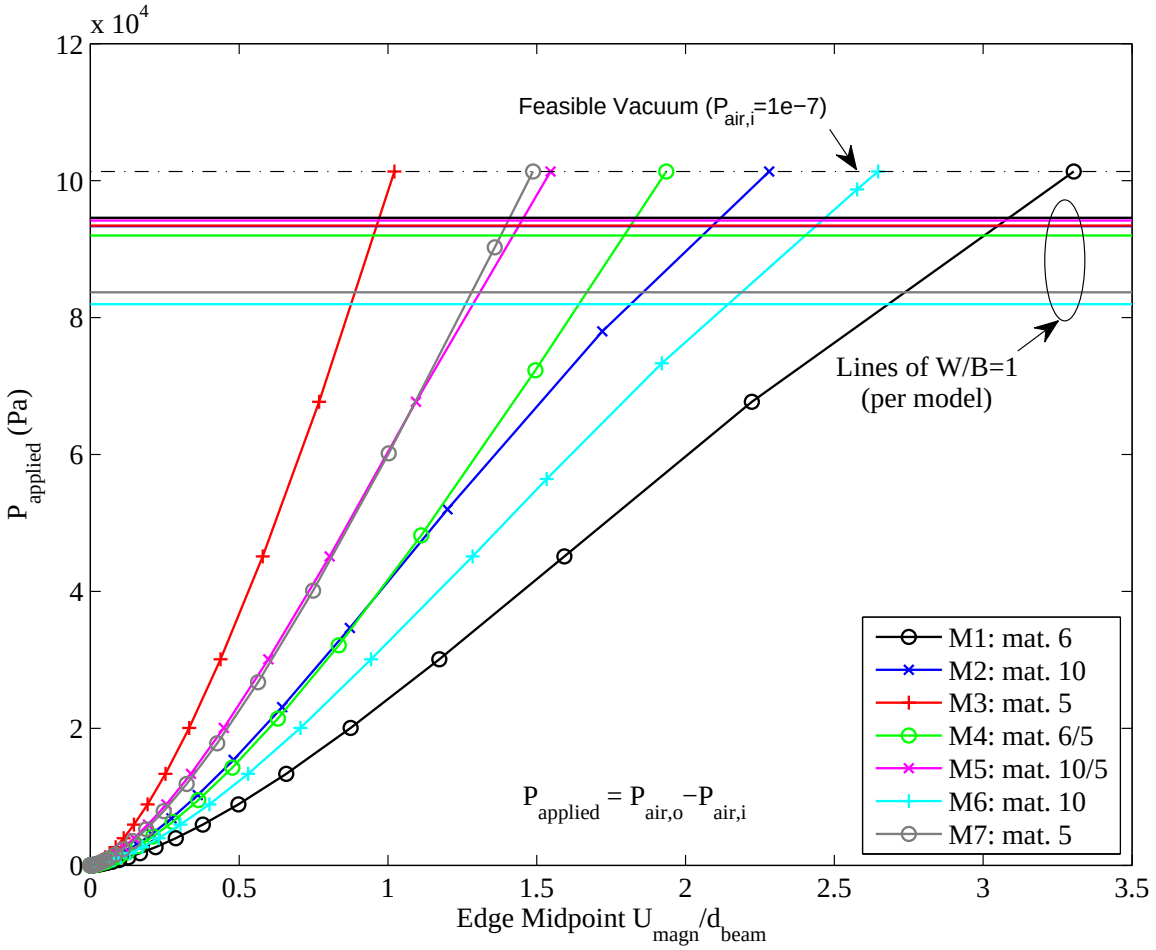


Figure 52: Icosahedron: Applied Pressure versus Edge Midpoint Displacement Normalized by the Beam's Diameter

are shown at the center of the beam, in the order of 1 to 3.3 times the beam's diameter. Note the nonlinear behavior for pressures less than 20 kPa (southwest corner of the graph); it indicates that the beam is acquiring stiffness, thus causing a change in slope.

However, the change in slope between Models 1 and 3 is 77%, lower than the change in slope when considering the vertex. The reduction in slope is an indicator of the model's capacity to sustain more load, as the bifurcation point occurs when the slope

goes to 0. These changes are visible in the upper side of models 2 and 7's curves, which both share the material properties of the Spectra Fiber. The Spectra fiber models are of the thickest ones since they have the lowest density, as a result of the set W/B. However, the Spectra fiber has the lowest modulus of elasticity of the three materials considered for evaluation. These results suggest that a stronger correlation to the modulus of elasticity exist than the one predicted by the specific stiffness index (proposed in Section 2.7).

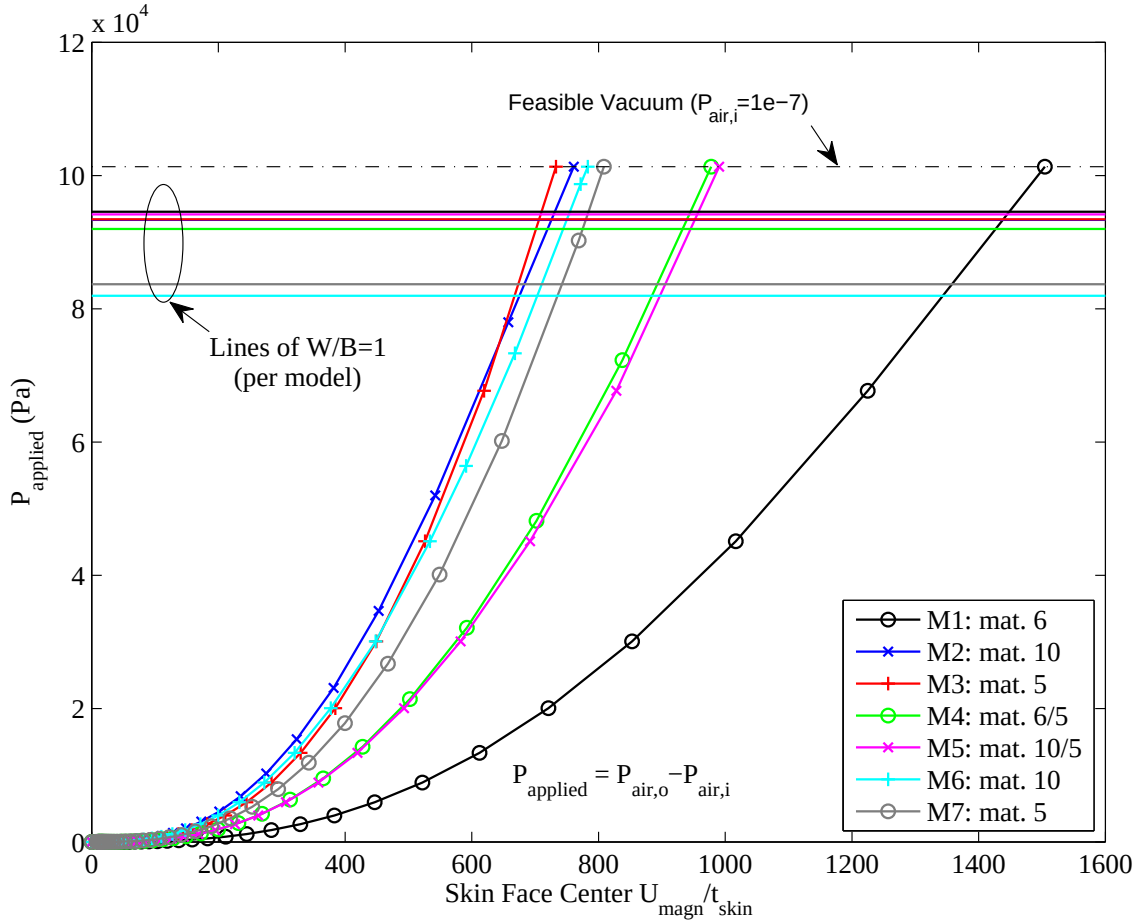


Figure 53: Icosahedron: Applied Pressure versus Normalized Skin Center Displacement

The third critical point is a triangular face center. The face center deflection, normalized by skin thickness as applied pressure increases, is shown in Figure 53 for all models. Note the nonlinearity in the entirety of the curves. The skin initially displaces considerably in the lower left corner, up to 200 times the skin thicknesses. This behavior is consistent with a membrane. Since there is no bending stiffness, the skin is required to deflect in order to acquire membrane stiffness. Note that the slope starts increasing significantly in all models after 200 times the thicknesses, a clear sign of a continuous increment in membrane forces.Hardening occurs as a result. Note the difference in slopes between model1, the less stiff model, and model 2. It is clear that specific stiffness does plays an important role in the overall stiffness of the structure, a desired result that minimizes buoyancy loss. When comparing models 2 and 3, a stiffer response would be expected in Model 3 since it is materially stiffer. But the density of model 2 is about 40% less, thus the skin thickness increases considerably for a desired W/B, ergo producing significant geometric stiffness that result in similar responses. These large deflections bring a numerical concern, the possibility of element distortion, which usually results in loss of accuracy. Therefore, the mesh was verified and no distortion was found; and the latter is believed to be a result of the mesh uniformity along the whole icosahedral structure.

This membrane stiffing behavior is of particular importance. In fact, the structural integrity of the icosahedron lies in this stiffing effect as much as in the stiffness that comes from the frame. Therefore, both frame and skin are dependent on each other to produce the overall stiffness. This phenomenon results from the selection of materials. If a considerably lower density is used in the skin, driving its thickness up, it would acquire

significant bending stiffness, changing the structural behavior and diminishing the frame's purpose to a point where the bending stiffness is so significant that the frame is no longer required. The issue here is finding that type of material.

At this point is known that large displacements occur in all parts of the icosahedron, the highest being the skin center. It is also known that those displacements have minimal effect on the W/B. Therefore all models appear to be feasible at this point. But, material failure as a result of stresses generated by those large displacements needs to be considered.

The critical points for maximum stress differ from those of displacement. In case of the frame, the maximum stress occurs at about 5% the beam's length measured from any vertex towards the beam's midpoint. Applied pressure versus maximum von Mises stress on the frame is plotted in Figure 54 for all models. Material failure lines, given by vertical dashed lines, represent, from left to right, the ultimate strength of material 6 (Beryllium), the ultimate strength of material 10 (Spectra fiber) and the yield strength of material 5 (CNT)[14]. Note that the stress behavior is very similar to the displacement behavior of the frame (see Figure 51). Failure lines demonstrate that models 1, 2, 4 and 6 fail before achieving buoyancy. Model 5 does not fail, but it gets very close to failure once the feasible vacuum line is reached, leaving no space for a safety factor. The stress distribution of five adjacent beams is shown in Figure 55 at maximum stress points in the direction along the beams. Beams show compression at interior and tension at the exterior, with an axial component, indicating that both axial and bending stress are present. Moving towards beams midpoints (towards the edges of the figure), the stress

[14]The yielding point of Carbon Nanotubes (CNT) is an approximation established by selecting the lowest value of the strength range given by the manufacturer. See Table 1 for more details.

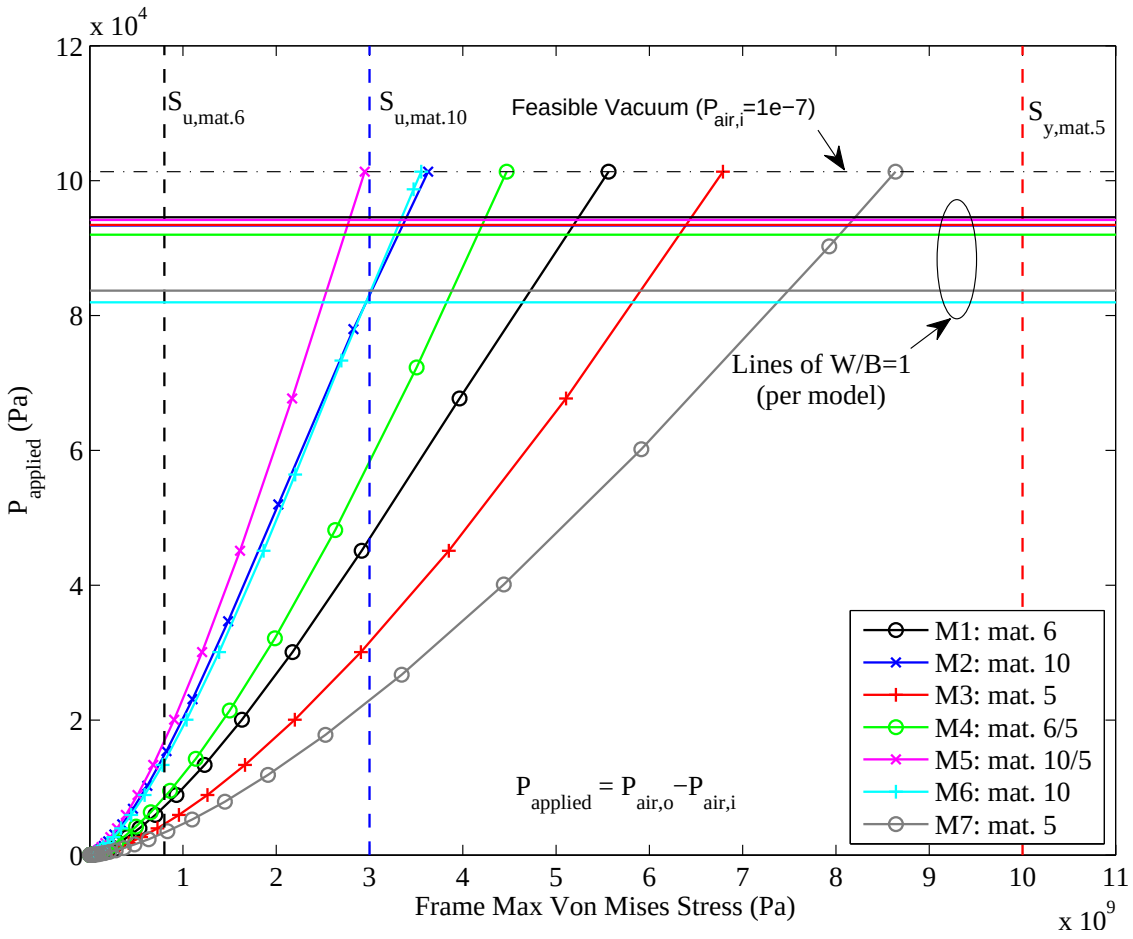


Figure 54: Icosahedron: Applied Pressure versus Frame Maximum von Mises Stress

dissipates by almost a magnitude. Therefore, the design of the joints becomes of importance in order to distribute stress uniformly and reduce maximum stress.

In the case of the skin, the stress critical points are at vertices or joints. Since vertices are modeled as points, stress concentrates around that area and creates a singularity. Those singularity points are shown in Figure 56. Note that the rest of the contour has considerably less stress and a uniform distribution. Therefore, two scenarios were considered in order to evaluate skin failure: with and without singularities. In order

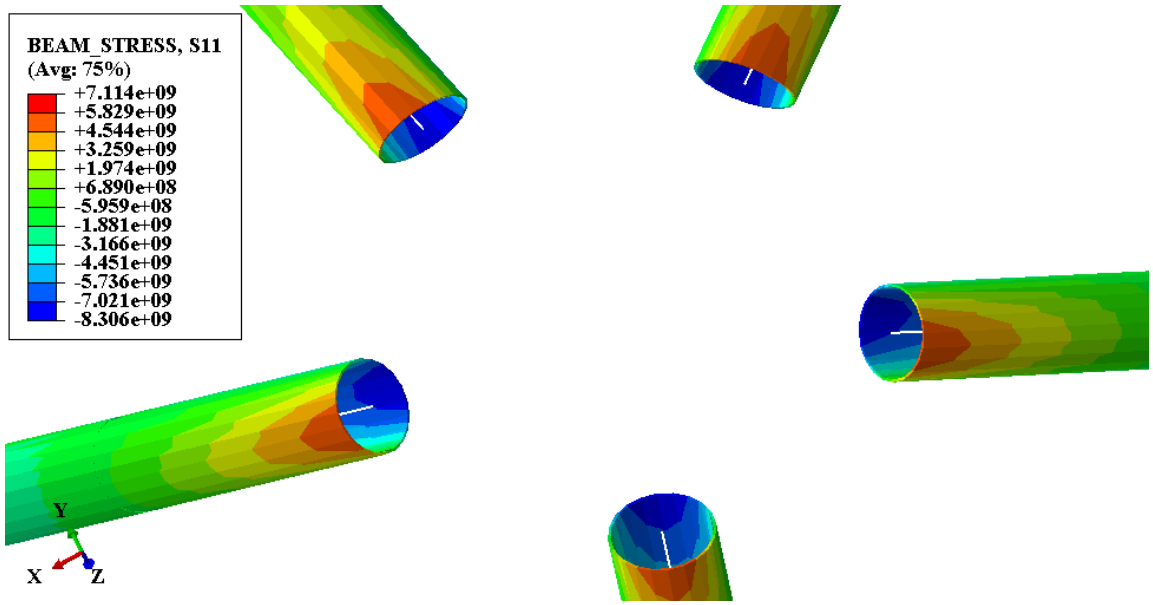


Figure 55: Icosahedron: Cross-sectional Stress Distribution of Five Adjacent Beams along their Axis

to eliminate the singularities, elements surrounding each vertex were eliminated for maximum stress calculation purposes. Applied pressure versus skin maximum stress with and without singularities are shown in Figure 57 and Figure 58, respectively. Contrary to skin displacement curves (Figure 53) where model 7 shows significantly less displacement, skin stress curves in Figure 57 show model 7 as the one with largest stress. This comes as a result of the selected W/B of 0.8 that drove the frame to small dimensions, ultimately causing greater stress on the skin. On the other hand, both models with the properties of Spectra Fiber (Table 1) -Models 2 and 6-, showed a third of the model 7 stress. This is a result of the increment in skin thickness.

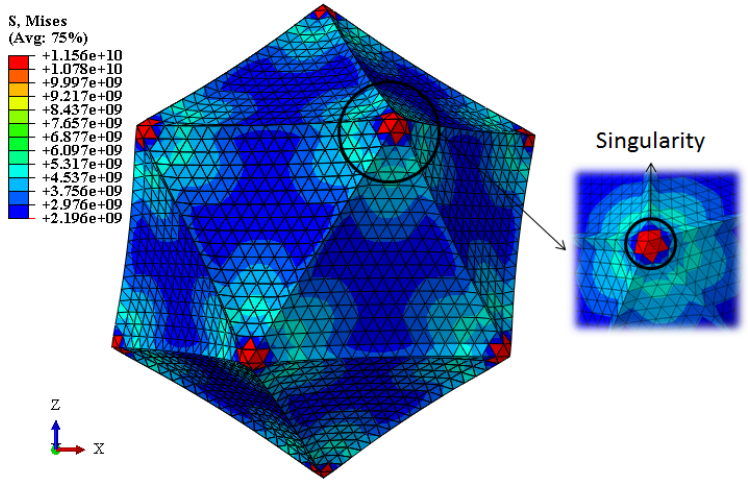


Figure 56: Icosahedron: von Mises Stress Contour of Model 3

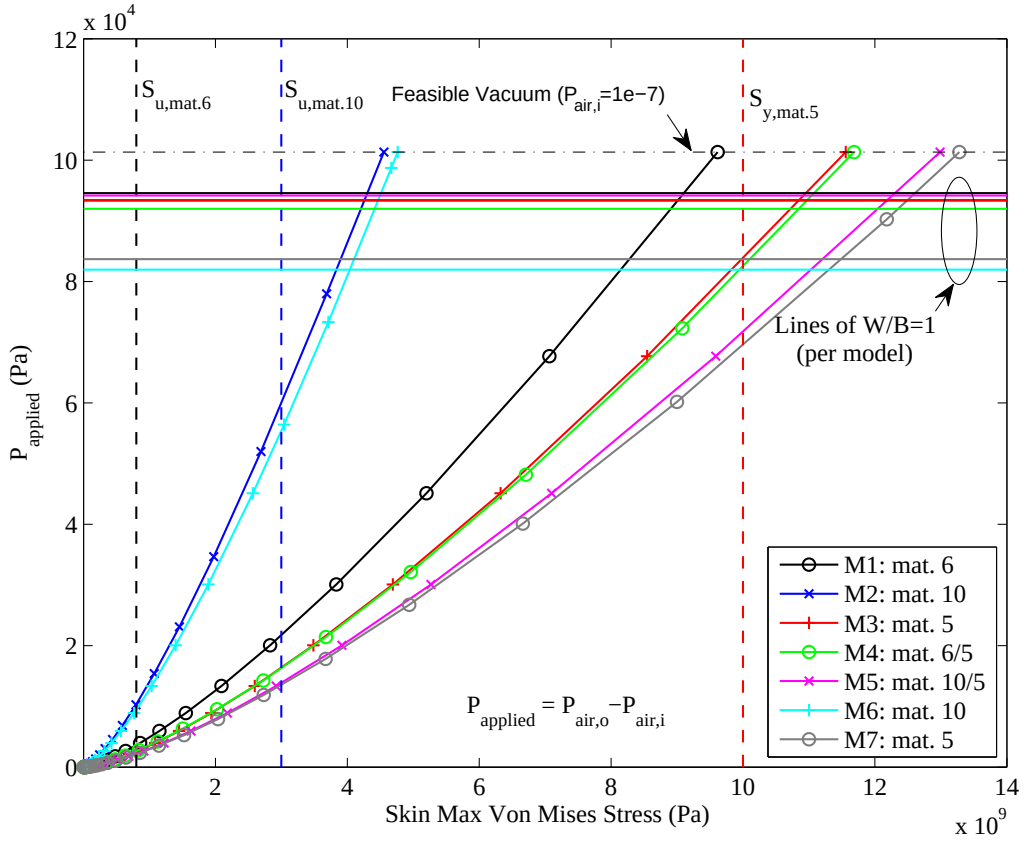


Figure 57: Icosahedron: Applied Pressure versus Skin Maximum von Mises Stress (with singularities)

But, without singularities (Figure 58), skin stress reduces considerably in all models. This is an indicator of the effect of singularities. Note that once singularities are not accounted for, only model 1 fails skin wise before achieving buoyancy. This bring an important modeling point: the connectivity of beams with surface elements can produce stress concentrations in the model that should not be in the real structure since it will most likely connect on a surface, not a point.

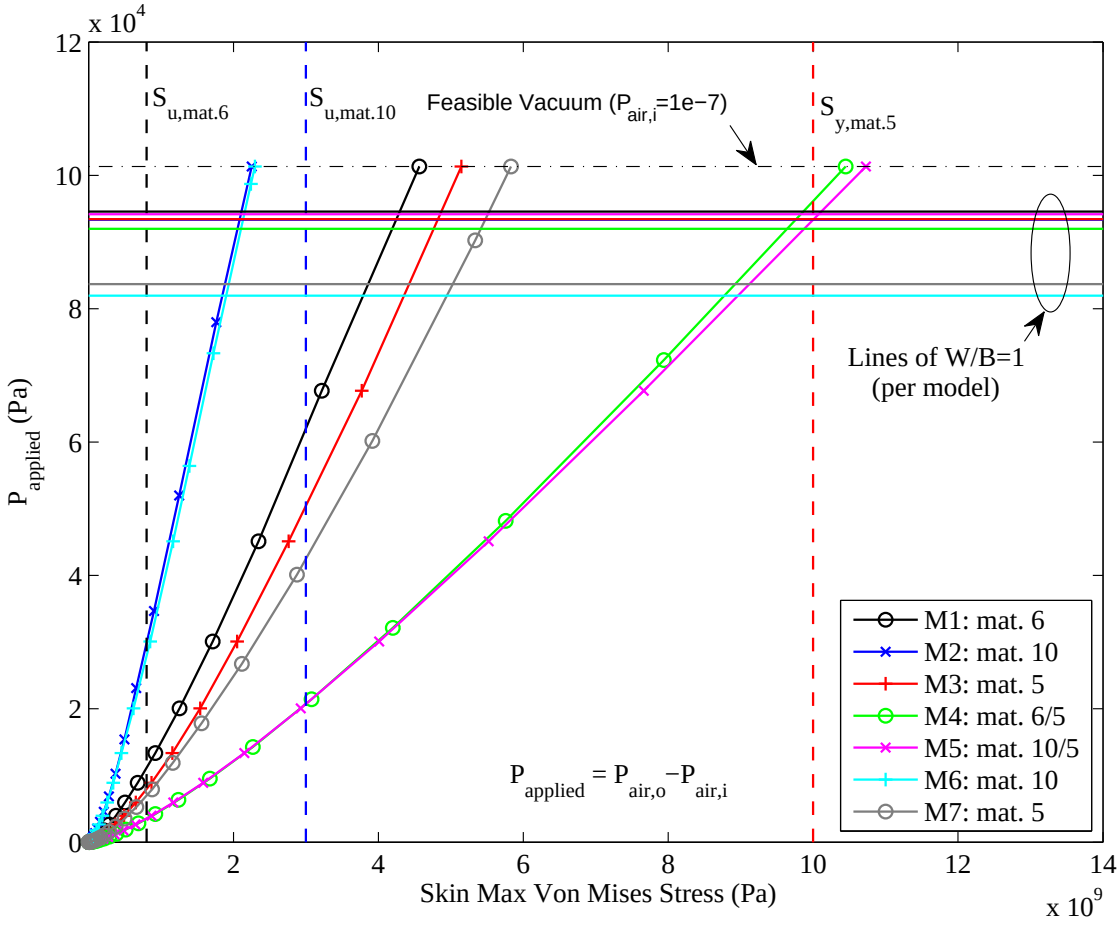


Figure 58: Icosahedron: Applied Pressure versus Skin Maximum von Mises Stress (without singularities)

Overall, the skin stress behaves fairly linear once applied pressure is more than 10 kPa, but note that stiffer models show significant slope increment, as previously seen in pressure versus displacement curves.

4.5 Summary

Before evaluating the structural response of the icosahedron, a symmetry validation was performed to ensure that the symmetric BC found in the frame's study (see Section 3.6.3) maintained symmetry throughout the analysis of the icosahedron. Results concluded that symmetry is indeed conserved and that the applied BC have virtually no effect in the structural response, as desired. Following this study, the results regarding a linear buckling analysis and nonlinear static analyses were presented.

The linear buckling analysis performed in model 3 estimated the buckling modes of the icosahedron, suggesting that the skin will buckle at pressure values as low as 7 Pa, while the frame remains rigid within the first 200 buckling modes. This was dimmed incorrect since: (1) the icosahedron displays nonlinear behavior (not captured by this analysis), (2) the use of a shell element was required since the membrane initially has no stiffness and the linear analysis is incapable of capturing such stiffness, resulting in compressive modes related to the small bending stiffness left in shell elements, undermining the skin response and the structure's buckling, and (3) the nonlinear analysis confirms the lack of buckling before the SL pressure is reached.

The nonlinear static analysis provided great insight on the structural response of the icosahedron. First, the volume reduction resulting from the skin deflection proved to be minimal, causing minimal effect on the W/B of the different models with the largest reduction being 0.04; consistent with what was found in the single triangle study (see

Section 3.5.3). Second, large displacements were found in all the parts of the icosahedron model, the least occurring at vertices, followed by edge midpoints, and the largest occurring at face centers. Regardless, all models remained stable during the entire analysis, with significant harding occurring in the skin that helped increase overall model's stiffness. Third, the stress proved to be the cause of failure for most models, just leaving models 3 and 7, both made entirely of CNT. Frame failure locations were at about 5% the beam's length from any vertex to any beam, with significantly higher values than the rest of the frame. This suggest that stiffening those areas not only will prevent failure, but it would make model 2 feasible. Results for models 3 and 7 are summarized in Table 8, including maximum displacements and stresses for both frame and skin, the W/B and the maximum altitude. The maximum altitude is predicated on the fact that an ultra high vacuum is used. The safety factors are calculated using the estimated yielding point.

Table 8: Feasible Models

Model:	**3**	**7**
r_{beam} (mm)	1.49	1.33
t_{skin} (mm)	0.0118	0.0118
W/B	0.91	0.81
Maximum Altitude - with Ultra High Vacuum (m : ft)	512 : 1680	676 : 2219
Material Properties		
Density (kg/m^3)	1650	1650
Poison's ratio	0.2	0.2
Modulus of Elasticity (GPa)	1000	1000
Frame		
Maximum Displacement (mm)	3.05	3.97
Maximum von Mises Stress (Pa)	6.79E+09	8.64E+09
Safety Factor	1.47	1.16
Skin		
Maximum Displacement (mm)	8.64	9.53
Maximum von Mises Stress (Pa)	9.62E+09	1.33E+10
Safety Factor (w.r.t the yielding point)	1.04	0.75
Maximum von Mises Stress - No Singularities (Pa)	5.15E+09	5.83E+09
Safety Factor - No Singularities (w.r.t the yielding point)	1.94	1.72

V. Conclusions and Recommendations

5.1 Chapter Overview

The research conducted in this thesis revolves around one question: what is the behavior of an LTA icosahedral structure subjected to a vacuum? Two tools were mainly used in order to answer such question: the Archimedes principle and FEA. Archimedes provides the principle of buoyancy, which puts the structure in the LTA realm. FEA provides the means of evaluating the response of such structure subjected to a vacuum. The complexity of the icosahedron in combination with (1) the numerical nature of the FEA and (2) the buoyancy principle led to multiple questions. These questions ultimately served to validate the techniques used to model its response and justify the use of the selected design features. The responses of these questions along with the understanding of the icosahedral structure are presented as conclusions in the next section. Following the conclusions, a research impact statement is provided and, recommendations are stated in order to provide a stepping stone for future research.

5.2 Conclusions of Research

The conclusions presented are divided into three categories: design, concepts and modeling techniques.

1. Design

    (a) The selection of an appropriate cross-section for the frame members greatly influences the stiffness and failure modes of the entire structure. The study showed that for beams of circular cross-section, the performance improves in

an exponential fashion as the thickness of the beams tend to 0. Therefore, manufacturability and material selection needs to be considered in order attain an improved performance.

(b) The material selection becomes the critical design factor for LTAV subjected to a vacuum. Preliminary studies show that the response is highly dependent on the specific stiffness, ergo the modulus of elasticity and density become the driving constraints with minimal stiffness effects from changes in the Poison's ratio. The response of the icosahedron shows material failure modes in some models, indicating the need of high specific strength in order to sustain the high stress levels that result from thin components.

(c) The membrane forces in the icosahedral skin provide significant stiffness to the overall structure. The skin shows significant hardening as a by product of the large deflections.

(d) The frame provides structural stability, allowing for the structure to sustain large deflections without collapsing.

2. Concepts

(a) Large displacements cause minimal changes in the W/B of the structure. The response of the different icosahedral models show skin displacements from 600 to 1500 times its thicknesses with only 2 to 4% in volume reduction. The magnitude of the volume reduction is a consequence of the fairly rigid frame, with maximum displacements in the order of 1 to 3.3 its diameter. Therefore, the W/B ratio was not affected by more than 0.04.

(b) Although a perfect vacuum is desired, a high partial vacuum can be used to achieve buoyancy. The vacuum level needed in order to achieve buoyancy depends on the design W/B, selected by assuming a full vacuum is achieved.

(c) An icosahedral LTAV has the potential of being constructed provided a high specific strength material is available. Results showed that the specific strength is the driving constraint and the cause of failure. Though the specific stiffness is important, it was shown that large displacements are tolerable with minimal effects on the structure's buoyancy. Furthermore, it was shown that a material with a strength of 30 GPa would allow a vehicle with a W/B of 0.82. Furthermore, if the frame beams are stiffened from the vertex up to 5% the beam's length, the strength requirement reduces considerably.

3. Modeling Techniques

(a) The Newton Raphson technique with adaptive automatic stabilization is an efficient analysis tool capable of performing nonlinear analyzes of initially flat membranes and frames, without the artificial damping added to stabilize the model affecting the solution's accuracy.

(b) The static analysis of the icosahedron model showed sensitivity to the provided initial load increment. Analyses of the icosahedral models showed convergence difficulties at the first load increment; which lead to the modification of the solution controls in Abaqus in order to achieve convergence.

(c) The shell element behaves as a membrane for thin enough surfaces. Preliminary studies showed that for a skin thickness $\leq 0.02\ mm$, the bending stiffness of the shell element becomes insignificant, ergo displaying membrane behavior. Nonetheless, the shell element underestimates the linear buckling characteristics of the icosahedron as a result of not accounting for the membrane stiffness.

(d) In order to preserve the symmetric characteristics of the icosahedron, proper BC need to be selected. Studies showed that fixing the displacement DOF on parallel planes of opposite vertices provides modeling symmetry through the analysis.

(e) The use of surface elements tied to beam elements produced modeling singularities that would not show in the real design. The skin showed significantly higher stress at the frame/skin vertices connection, creating singularity points. In the real design, the frame and skin would meet at a surface rather than a point within the vertices, eliminating those singularity points.

5.3 Research Impact

This research has two areas of impact: the nonlinear structural analysis of an icosahedron and its applicability to lighter-than-air vehicles (LTAV) subjected to a vacuum. Typical literature offers a great amount of background on the structural behavior of simple geometries, but they tend to lack the background on complex structures, particularly the icosahedron. The structural response of a complex geometry such as the icosahedron is a unique problem that relies on nonlinear theories and numerical methods

in order to understand the behavior of such a structure. Additionally, its applicability to LTAV not only provides a ground of measure, but it establishes the capacity of such structure to achieve buoyancy. Therefore, the largest contribution of this research is the background on the nonlinear response of the icosahedral structure and its vacuum LTAV's potential.

5.4 Recommendations for Future Research

The following recommendations extend from the lessons learned during the research process and the limitations that the modeling techniques presented:

1. The icosahedral skin and frame were modeled using beam and surface elements, respectively, providing an efficient analysis technique. As a result, the skin showed significant stress concentrations at the icosahedral vertices where the beam and membrane elements connect at a point rather than a surface, creating a singularity. A three dimensional analysis would provide insight on the effect of these singularities and the appropriate design of the vertices.

2. The Newton Raphson with adaptive automatic stabilization showed to be an accurate analysis technique that captures the nonlinearities of membrane and beams. On the other hand, this technique does not have the capacity of modeling the post buckling response. A dynamic explicit analysis is recommended in order to capture the post-buckling response of the icosahedral structure and the dynamic behavior that comes about the structure being vacated to produce buoyancy. The use of imperfections in the buckling analysis, such as localized design deviations,

is recommended in order to capture the structure's vulnerability to the different imperfections introduced during the manufacturing process.

3. A study of the effects that other beam cross-sections have in the overall stiffness and instability of the icosahedron is recommended, in order to minimize the weight and maximize the stiffness and strength of the frame.

4. The analysis conducted in this research presumed that the material behaved linearly, an assumption that serves well as a first approximation, when the material properties required for proper structural response are not known. The need of materials with high specific stiffness and strength drives the material selection to composite type materials, materials that more often than not, respond nonlinearly. Therefore, the inclusion of nonlinear effects along with the geometric nonlinearities is recommended.

5. The research presented here used an icosahedral structure of fixed diameter, neglecting possible effects brought by changes in size. Therefore, it is recommended to evaluate the possible effects on the structural response that changes in dimensionality can bring.

6. Other considerations that need to be taken into account include: skin diffusivity, aerodynamic effects, propulsion and manufacturability.

Appendix A: Modeling Studies Tabulated Results

A.1 Square Membrane Convergence Study Results

The following table shows the results from the square membrane convergence study. The model properties are: modulus of elasticity (303 GPa), Poison's ratio (0.3), membrane thickness (0.05 mm). The Newton Raphson with adaptive automatic stabilization technique is used for the FEA. *Edge Seed* refers to the amount of elements along each edge and *# Elements* is the resulting number of elements. *S3R*, *M3D3*, *Timoshenko* and *Seide* refers to the out of plane center displacement of the shell element, membrane element, Timoshenko's solution (see Equation (2.32) and Equation (2.33)) and Seide's solution (see Equation (2.35)), respectively.

Table 9: Square Membrane Convergence Study Results

		U_3 (*out of plane center displacement*) (*m*)				% Error			
Edge Seed	# Elements	S3R	M3D3	Timoshenko	Seide	S3R versus Timoshenko	M3D3 versus Timoshenko	S3R vs Seide	M3D3 versus Seide
8	128	-2.666E-03	-2.674E-03	-2.991E-03	-2.690E-03	10.89%	10.59%	0.91%	0.58%
9	162	-2.621E-03	-2.626E-03	-2.991E-03	-2.690E-03	12.37%	12.19%	2.55%	2.36%
10	200	-2.671E-03	-2.681E-03	-2.991E-03	-2.690E-03	10.69%	10.38%	0.69%	0.34%
11	242	-2.646E-03	-2.648E-03	-2.991E-03	-2.690E-03	11.53%	11.47%	1.63%	1.56%
12	288	-2.683E-03	-2.685E-03	-2.991E-03	-2.690E-03	10.29%	10.25%	0.24%	0.20%
13	338	-2.661E-03	-2.661E-03	-2.991E-03	-2.690E-03	11.03%	11.05%	1.06%	1.09%
14	392	-2.685E-03	-2.687E-03	-2.991E-03	-2.690E-03	10.23%	10.17%	0.18%	0.12%
15	450	-2.669E-03	-2.669E-03	-2.991E-03	-2.690E-03	10.77%	10.78%	0.77%	0.79%
16	512	-2.688E-03	-2.688E-03	-2.991E-03	-2.690E-03	10.12%	10.12%	0.06%	0.06%
17	578	-2.675E-03	-2.674E-03	-2.991E-03	-2.690E-03	10.59%	10.60%	0.57%	0.59%
18	648	-2.689E-03	-2.690E-03	-2.991E-03	-2.690E-03	10.10%	10.08%	0.03%	0.01%
19	722	-2.678E-03	-2.678E-03	-2.991E-03	-2.690E-03	10.46%	10.46%	0.44%	0.44%
20	800	-2.690E-03	-2.690E-03	-2.991E-03	-2.690E-03	10.06%	10.05%	0.02%	0.02%
21	882	-2.681E-03	-2.681E-03	-2.991E-03	-2.690E-03	10.37%	10.37%	0.33%	0.33%

Continued on next page

Table 9 – continued from previous page

		U3 (vertical displacement) (m)				% Error			
Edge Seed	# Elements	S3R	M3D3	Timoshenko	Seide	S3R versus Timoshenko	M3D3 versus Timoshenko	S3R vs Seide	M3D3 versus Seide
22	968	-2.691E-03	-2.691E-03	-2.991E-03	-2.690E-03	10.03%	10.03%	0.04%	0.04%
23	1058	-2.683E-03	-2.683E-03	-2.991E-03	-2.690E-03	10.30%	10.30%	0.25%	0.25%
24	1152	-2.692E-03	-2.692E-03	-2.991E-03	-2.690E-03	10.02%	10.02%	0.06%	0.06%
25	1250	-2.685E-03	-2.685E-03	-2.991E-03	-2.690E-03	10.24%	10.24%	0.19%	0.19%
26	1352	-2.692E-03	-2.692E-03	-2.991E-03	-2.690E-03	10.01%	10.01%	0.07%	0.07%
27	1458	-2.686E-03	-2.686E-03	-2.991E-03	-2.690E-03	10.20%	10.20%	0.14%	0.14%
28	1568	-2.692E-03	-2.692E-03	-2.991E-03	-2.690E-03	10.00%	10.00%	0.08%	0.08%
29	1682	-2.687E-03	-2.687E-03	-2.991E-03	-2.690E-03	10.16%	10.16%	0.10%	0.11%
30	1800	-2.692E-03	-2.692E-03	-2.991E-03	-2.690E-03	9.99%	9.99%	0.09%	0.09%
31	1922	-2.688E-03	-2.688E-03	-2.991E-03	-2.690E-03	10.14%	10.14%	0.07%	0.07%
32	2048	-2.693E-03	-2.693E-03	-2.991E-03	-2.690E-03	9.98%	9.98%	0.10%	0.10%
33	2178	-2.689E-03	-2.689E-03	-2.991E-03	-2.690E-03	10.11%	10.11%	0.05%	0.05%
34	2312	-2.693E-03	-2.693E-03	-2.991E-03	-2.690E-03	9.98%	9.98%	0.10%	0.10%
35	2450	-2.689E-03	-2.689E-03	-2.991E-03	-2.690E-03	10.09%	10.09%	0.03%	0.03%
36	2592	-2.693E-03	-2.693E-03	-2.991E-03	-2.690E-03	9.97%	9.97%	0.11%	0.11%
37	2738	-2.690E-03	-2.690E-03	-2.991E-03	-2.690E-03	10.08%	10.08%	0.01%	0.01%
38	2888	-2.693E-03	-2.693E-03	-2.991E-03	-2.690E-03	9.97%	9.97%	0.11%	0.11%
39	3042	-2.690E-03	-2.690E-03	-2.991E-03	-2.690E-03	10.06%	10.06%	0.01%	0.01%
40	3200	-2.693E-03	-2.693E-03	-2.991E-03	-2.690E-03	9.97%	9.97%	0.12%	0.11%
41	3362	-2.691E-03	-2.691E-03	-2.991E-03	-2.690E-03	10.05%	10.05%	0.02%	0.02%
42	3528	-2.693E-03	-2.693E-03	-2.991E-03	-2.690E-03	9.96%	9.97%	0.12%	0.12%
43	3698	-2.691E-03	-2.691E-03	-2.991E-03	-2.690E-03	10.04%	10.04%	0.03%	0.03%

A.2 Circular Membrane Convergence Study Results

The following table shows the results from the circular membrane convergence study. The model properties are: modulus of elasticity (303 GPa), Poison's ratio (0.25), membrane thickness (1 mm) and M3D3 membrane elements. The Newton Raphson with adaptive automatic stabilization technique is used for the FEA. **Edge Seed** refers to the amount of elements at the edge. $U_{3\,FEA}$ and $U_{3\,Analytical}$ refers to the FEA and analytical (see Equation (2.29)) out of plane center displacements, respectively. S_{FEA} and $S_{Analytical}$ refers to the FEA von Mises and analytical (see Equation (2.30)) stress, respectively.

Table 10: Circular Membrane Convergence Study Results

Edge Seed	# Elements	$U_{3\,FEA}$ (m)	$U_{3\,Analytical}$ (m)	U_3 % Error	S_{FEA} (Pa)	$S_{Analytical}$ (Pa)	S % Error
5	5	-1.0392E-03	-1.0671E-03	2.6%	6.79E+07	9.41E+07	27.9%
6	6	-1.1379E-03	-1.0671E-03	6.6%	7.10E+07	9.41E+07	24.5%
7	7	-1.1996E-03	-1.0671E-03	12.4%	7.29E+07	9.41E+07	22.5%
8	12	-9.5939E-04	-1.0671E-03	10.1%	7.78E+07	9.41E+07	17.3%
9	13	-9.5252E-04	-1.0671E-03	10.7%	7.90E+07	9.41E+07	16.0%
10	16	-9.6707E-04	-1.0671E-03	9.4%	7.87E+07	9.41E+07	16.4%
11	19	-9.7388E-04	-1.0671E-03	8.7%	7.99E+07	9.41E+07	15.1%
12	24	-1.0986E-03	-1.0671E-03	3.0%	8.31E+07	9.41E+07	11.7%
13	27	-1.0917E-03	-1.0671E-03	2.3%	8.34E+07	9.41E+07	11.4%
14	32	-1.0586E-03	-1.0671E-03	0.8%	8.67E+07	9.41E+07	7.9%
15	37	-1.0278E-03	-1.0671E-03	3.7%	8.88E+07	9.41E+07	5.6%
16	42	-1.0468E-03	-1.0671E-03	1.9%	8.92E+07	9.41E+07	5.2%
17	45	-1.0600E-03	-1.0671E-03	0.7%	8.97E+07	9.41E+07	4.6%
18	50	-1.0714E-03	-1.0671E-03	0.4%	8.94E+07	9.41E+07	5.0%
19	71	-1.0496E-03	-1.0671E-03	1.6%	9.11E+07	9.41E+07	3.2%
20	74	-1.0509E-03	-1.0671E-03	1.5%	9.11E+07	9.41E+07	3.2%
21	79	-1.0503E-03	-1.0671E-03	1.6%	9.16E+07	9.41E+07	2.6%
22	88	-1.0592E-03	-1.0671E-03	0.7%	9.17E+07	9.41E+07	2.6%
23	99	-1.0605E-03	-1.0671E-03	0.6%	9.17E+07	9.41E+07	2.6%
24	108	-1.0781E-03	-1.0671E-03	1.0%	9.18E+07	9.41E+07	2.5%
25	117	-1.0594E-03	-1.0671E-03	0.7%	9.29E+07	9.41E+07	1.3%
26	126	-1.0576E-03	-1.0671E-03	0.9%	9.27E+07	9.41E+07	1.5%
27	133	-1.0579E-03	-1.0671E-03	0.9%	9.27E+07	9.41E+07	1.5%

Continued on next page

Table 10 – continued from previous page

Edge Seed	# Elements	$U_{3\ FEA}\ (m)$	$U_{3\ Analytical}\ (m)$	$U_3\ \%\ Error$	$S_{FEA}\ (Pa)$	$S_{Analytical}\ (Pa)$	$S\ \%\ Error$
28	142	-1.0709E-03	-1.0671E-03	0.4%	9.30E+07	9.41E+07	1.2%
29	151	-1.0678E-03	-1.0671E-03	0.1%	9.27E+07	9.41E+07	1.5%
30	172	-1.0645E-03	-1.0671E-03	0.2%	9.36E+07	9.41E+07	0.6%
31	177	-1.0640E-03	-1.0671E-03	0.3%	9.34E+07	9.41E+07	0.7%
32	186	-1.0668E-03	-1.0671E-03	0.0%	9.35E+07	9.41E+07	0.6%
33	199	-1.0652E-03	-1.0671E-03	0.2%	9.34E+07	9.41E+07	0.8%
34	216	-1.0749E-03	-1.0671E-03	0.7%	9.33E+07	9.41E+07	0.8%
35	223	-1.0658E-03	-1.0671E-03	0.1%	9.38E+07	9.41E+07	0.4%
36	242	-1.0661E-03	-1.0671E-03	0.1%	9.38E+07	9.41E+07	0.3%
37	249	-1.0679E-03	-1.0671E-03	0.1%	9.38E+07	9.41E+07	0.3%
38	272	-1.0675E-03	-1.0671E-03	0.0%	9.37E+07	9.41E+07	0.4%
39	285	-1.0715E-03	-1.0671E-03	0.4%	9.36E+07	9.41E+07	0.5%
40	298	-1.0681E-03	-1.0671E-03	0.1%	9.40E+07	9.41E+07	0.1%
41	305	-1.0681E-03	-1.0671E-03	0.1%	9.40E+07	9.41E+07	0.1%
42	330	-1.0673E-03	-1.0671E-03	0.0%	9.40E+07	9.41E+07	0.1%
43	335	-1.0693E-03	-1.0671E-03	0.2%	9.40E+07	9.41E+07	0.1%
44	348	-1.0713E-03	-1.0671E-03	0.4%	9.40E+07	9.41E+07	0.1%
45	363	-1.0685E-03	-1.0671E-03	0.1%	9.41E+07	9.41E+07	0.0%
46	382	-1.0705E-03	-1.0671E-03	0.3%	9.41E+07	9.41E+07	0.0%
47	399	-1.0693E-03	-1.0671E-03	0.2%	9.41E+07	9.41E+07	0.0%
48	414	-1.0699E-03	-1.0671E-03	0.3%	9.42E+07	9.41E+07	0.1%
49	445	-1.0723E-03	-1.0671E-03	0.5%	9.41E+07	9.41E+07	0.0%
50	452	-1.0706E-03	-1.0671E-03	0.3%	9.42E+07	9.41E+07	0.1%
51	475	-1.0736E-03	-1.0671E-03	0.6%	9.40E+07	9.41E+07	0.1%

A.3 Circular Membrane Thickness Study Results

The following table shows the results from the circular membrane thickness study, comparing the FE solution with the analytical solution (see Equation (2.29) and Equation (2.30)). The model properties are: modulus of elasticity (303 GPa), Poison's ratio (0.25) and 452 S3R shell elements. The Newton Raphson with adaptive automatic stabilization technique is used for the FEA. $\boldsymbol{t}$ refers to the membrane thickness. $\boldsymbol{U_3}$ and $\boldsymbol{S}$ refer to the center out-of-plane displacement and stress, respectively. In case of the FE results, stresses are given at the bottom, middle and top cross-sectional points. The stress % difference is based of the shell middle stress, $\boldsymbol{S_{middle}}$, and the analytical stress, $\boldsymbol{S}$.

Table 11: Circular Membrane Thickness Study Results

$t\ (m)$	Analytical Membrane		S3R Shell Element				% Difference	
	$U_3\ (m)$	$S\ (Pa)$	$U_3\ (m)$	$S_{bottom}\ (Pa)$	$S_{middle}\ (Pa)$	$S_{top}\ (Pa)$	$U_3\ (m)$	S
5.00E-06	-6.24E-03	3.22E+09	-6.02E-03	3.37E+09	3.37E+09	3.36E+09	3.59%	4.61%
6.00E-06	-5.87E-03	2.85E+09	-5.66E-03	2.98E+09	2.98E+09	2.97E+09	3.68%	4.49%
7.00E-06	-5.58E-03	2.57E+09	-5.37E-03	2.69E+09	2.68E+09	2.68E+09	3.74%	4.39%
8.00E-06	-5.34E-03	2.35E+09	-5.13E-03	2.46E+09	2.45E+09	2.45E+09	3.78%	4.32%
9.00E-06	-5.13E-03	2.17E+09	-4.93E-03	2.27E+09	2.27E+09	2.26E+09	3.82%	4.26%
1.00E-05	-4.95E-03	2.03E+09	-4.76E-03	2.12E+09	2.11E+09	2.11E+09	3.86%	4.21%
2.00E-05	-3.93E-03	1.28E+09	-3.77E-03	1.34E+09	1.33E+09	1.32E+09	4.03%	3.96%
3.00E-05	-3.43E-03	9.75E+08	-3.29E-03	1.02E+09	1.01E+09	1.00E+09	4.10%	3.85%
4.00E-05	-3.12E-03	8.05E+08	-2.99E-03	8.49E+08	8.35E+08	8.21E+08	4.13%	3.79%
5.00E-05	-2.90E-03	6.93E+08	-2.78E-03	7.36E+08	7.19E+08	7.03E+08	4.16%	3.75%
6.00E-05	-2.73E-03	6.14E+08	-2.61E-03	6.56E+08	6.37E+08	6.18E+08	4.17%	3.72%
7.00E-05	-2.59E-03	5.54E+08	-2.48E-03	5.95E+08	5.74E+08	5.54E+08	4.18%	3.69%
8.00E-05	-2.48E-03	5.07E+08	-2.37E-03	5.48E+08	5.25E+08	5.03E+08	4.19%	3.67%
9.00E-05	-2.38E-03	4.69E+08	-2.28E-03	5.10E+08	4.86E+08	4.61E+08	4.19%	3.64%
1.00E-04	-2.30E-03	4.37E+08	-2.20E-03	4.79E+08	4.53E+08	4.26E+08	4.19%	3.62%
2.00E-04	-1.82E-03	2.75E+08	-1.75E-03	3.27E+08	2.84E+08	2.42E+08	4.12%	3.33%
3.00E-04	-1.59E-03	2.10E+08	-1.53E-03	2.72E+08	2.16E+08	1.60E+08	3.98%	2.77%
4.00E-04	-1.45E-03	1.73E+08	-1.39E-03	2.46E+08	1.77E+08	1.07E+08	3.85%	1.85%
5.00E-04	-1.34E-03	1.49E+08	-1.29E-03	2.33E+08	1.50E+08	6.75E+07	3.85%	0.45%
6.00E-04	-1.27E-03	1.32E+08	-1.21E-03	2.27E+08	1.30E+08	3.36E+07	4.17%	1.60%
7.00E-04	-1.20E-03	1.19E+08	-1.14E-03	2.25E+08	1.14E+08	3.23E+06	4.99%	4.51%

Continued on next page

Table 11 – continued from previous page

$t\ (m)$	Analytical Membrane		S3R Shell Element				% Difference	
	$U_3\ (m)$	$S\ (Pa)$	$U_3\ (m)$	$S_{bottom}\ (Pa)$	$S_{middle}\ (Pa)$	$S_{top}\ (Pa)$	$U_3\ (m)$	S
8.00E-04	-1.15E-03	1.09E+08	-1.08E-03	2.24E+08	1.00E+08	2.43E+07	6.44%	8.40%
9.00E-04	-1.11E-03	1.01E+08	-1.01E-03	2.24E+08	8.75E+07	4.88E+07	8.61%	13.35%
1.00E-03	-1.07E-03	9.41E+07	-9.44E-04	2.22E+08	7.59E+07	7.00E+07	11.53%	19.34%
2.00E-03	-8.47E-04	5.93E+07	-3.14E-04	1.15E+08	8.23E+06	9.87E+07	62.95%	86.12%
3.00E-03	-7.40E-04	4.52E+07	-9.73E-05	5.06E+07	7.90E+05	4.90E+07	86.85%	98.25%
4.00E-03	-6.72E-04	3.73E+07	-4.12E-05	2.82E+07	1.42E+05	2.79E+07	93.87%	99.62%
5.00E-03	-6.24E-04	3.22E+07	-2.12E-05	1.80E+07	3.74E+04	1.79E+07	96.61%	99.88%
6.00E-03	-5.87E-04	2.85E+07	-1.23E-05	1.25E+07	1.26E+04	1.25E+07	97.90%	99.96%
7.00E-03	-5.58E-04	2.57E+07	-7.79E-06	9.17E+06	5.05E+03	9.16E+06	98.60%	99.98%
8.00E-03	-5.34E-04	2.35E+07	-5.25E-06	7.02E+06	2.29E+03	7.02E+06	99.02%	99.99%
9.00E-03	-5.13E-04	2.17E+07	-3.71E-06	5.55E+06	1.14E+03	5.55E+06	99.28%	99.99%
1.00E-02	-4.95E-04	2.03E+07	-2.72E-06	4.49E+06	6.16E+02	4.49E+06	99.45%	100.00%
2.00E-02	-3.93E-04	1.28E+07	-3.77E-07	1.12E+06	1.12E+01	1.12E+06	99.90%	100.00%
3.00E-02	-3.43E-04	9.75E+06	-1.30E-07	4.99E+05	1.21E+00	4.99E+05	99.96%	100.00%
4.00E-02	-3.12E-04	8.05E+06	-6.52E-08	2.81E+05	3.22E-01	2.81E+05	99.98%	100.00%
5.00E-02	-2.90E-04	6.93E+06	-4.04E-08	1.80E+05	1.30E-01	1.80E+05	99.99%	100.00%
6.00E-02	-2.73E-04	6.14E+06	-2.83E-08	1.25E+05	6.80E-02	1.25E+05	99.99%	100.00%
7.00E-02	-2.59E-04	5.54E+06	-2.15E-08	9.16E+04	4.10E-02	9.16E+04	99.99%	100.00%
8.00E-02	-2.48E-04	5.07E+06	-1.72E-08	7.02E+04	2.80E-02	7.02E+04	99.99%	100.00%
9.00E-02	-2.38E-04	4.69E+06	-1.44E-08	5.54E+04	2.00E-02	5.54E+04	99.99%	100.00%
1.00E-01	-2.30E-04	4.37E+06	-1.23E-08	4.49E+04	1.50E-02	4.49E+04	99.99%	100.00%
2.00E-01	-1.82E-04	2.75E+06	-5.17E-09	1.12E+04	3.00E-03	1.12E+04	100.00%	100.00%

A.4 Triangular Membrane Convergence Study

The following table shows the results from the triangular membrane convergence study, comparing the relative change in displacement and stress as the number of element increases. The Newton Raphson with adaptive automatic stabilization technique is used for the FEA.The model properties are: modulus of elasticity (303 GPa), Poison's ratio (0.25) and membrane thickness (1 mm). **Edge Seed** refers to the amount of elements along each edge and **# Elements** is the resulting number of elements. $\boldsymbol{U_3}$ and $\boldsymbol{U_{3\,Diff}}$ are the magnitude and % error of the center out-of-plane displacement. $\boldsymbol{S}$ and $\boldsymbol{S_{Diff}}$ are the magnitude and % error of the von Mises stress. Two error parameters are considered. First, the $\boldsymbol{i}$ & $\boldsymbol{i-1}$ error considers the row i versus the row $i-1$. Second, the $\boldsymbol{i}$ & $\boldsymbol{i_{end}}$ error considers the last row versus the row i.

Table 12: Triangular Membrane Convergence Study Results

Edge Seed	# Elements	U_3 (m)	$U_{3\,Diff}$: i & $i-1$	$U_{3\,Diff}$: i & i_{end}	S (Pa)	S_{Diff} : i & $i-1$	S_{Diff} : i & i_{end}
5	25	-7.73E-04	-	2.7%	9.28E+07	-	3.1%
6	36	-8.86E-04	13.6%	0.7%	8.04E+07	14.4%	0.5%
7	49	-8.33E-04	6.2%	0.9%	7.15E+07	11.6%	3.4%
8	64	-8.30E-04	0.3%	0.9%	9.08E+07	23.8%	2.5%
9	81	-8.72E-04	5.0%	0.3%	8.18E+07	10.5%	0.1%
10	100	-8.46E-04	3.0%	0.4%	7.49E+07	8.7%	2.3%
11	121	-8.45E-04	0.1%	0.5%	8.91E+07	17.2%	2.0%
12	144	-8.68E-04	2.6%	0.2%	8.22E+07	8.0%	0.0%
13	169	-8.52E-04	1.8%	0.3%	7.67E+07	6.9%	1.7%
14	196	-8.52E-04	0.0%	0.3%	8.79E+07	13.5%	1.7%
15	225	-8.65E-04	1.6%	0.1%	8.24E+07	6.4%	0.1%
16	256	-8.55E-04	1.2%	0.2%	7.79E+07	5.7%	1.3%
17	289	-8.55E-04	0.0%	0.2%	8.71E+07	11.1%	1.5%
18	324	-8.64E-04	1.1%	0.1%	8.26E+07	5.3%	0.1%
19	361	-8.57E-04	0.8%	0.1%	7.87E+07	4.8%	1.1%
20	400	-8.57E-04	0.0%	0.1%	8.65E+07	9.4%	1.3%
21	441	-8.64E-04	0.8%	0.1%	8.26E+07	4.5%	0.2%

Continued on next page

Table 12 – continued from previous page

Edge Seed	# Elements	U_3 (m)	$U_{3\,Diff}$: i & $i-1$	$U_{3\,Diff}$: i & i_{end}	S (Pa)	S_{Diff} : i & $i-1$	S_{Diff} : i & i_{end}
22	484	-8.58E-04	0.6%	0.1%	7.93E+07	4.2%	0.9%
23	529	-8.58E-04	0.0%	0.1%	8.60E+07	8.2%	1.2%
24	576	-8.63E-04	0.6%	0.0%	8.27E+07	4.0%	0.2%
25	625	-8.59E-04	0.5%	0.1%	7.97E+07	3.7%	0.7%
26	676	-8.59E-04	0.0%	0.1%	8.57E+07	7.2%	1.1%
27	729	-8.63E-04	0.5%	0.0%	8.27E+07	3.5%	0.2%
28	784	-8.60E-04	0.4%	0.1%	8.00E+07	3.3%	0.6%
29	841	-8.60E-04	0.0%	0.1%	8.54E+07	6.5%	1.0%
30	900	-8.63E-04	0.4%	0.0%	8.27E+07	3.2%	0.2%
31	961	-8.60E-04	0.3%	0.0%	8.03E+07	3.0%	0.6%
32	1024	-8.60E-04	0.0%	0.0%	8.51E+07	5.9%	0.9%
33	1089	-8.62E-04	0.3%	0.0%	8.27E+07	2.9%	0.2%
34	1156	-8.60E-04	0.3%	0.0%	8.05E+07	2.7%	0.5%
35	1225	-8.60E-04	0.0%	0.0%	8.50E+07	5.4%	0.9%
36	1296	-8.62E-04	0.2%	0.0%	8.28E+07	2.6%	0.2%
37	1369	-8.60E-04	0.2%	0.0%	8.07E+07	2.5%	0.4%
38	1312	-8.60E-04	0.0%	0.0%	8.40E+07	4.0%	0.6%
39	1383	-8.61E-04	0.1%	0.0%	8.32E+07	1.0%	0.3%
40	1470	-8.62E-04	0.1%	0.0%	8.30E+07	0.2%	0.3%
41	1551	-8.60E-04	0.1%	0.0%	8.13E+07	2.0%	0.2%
42	1624	-8.61E-04	0.1%	0.0%	8.24E+07	1.3%	0.1%
43	1689	-8.61E-04	0.0%	0.0%	8.26E+07	0.3%	0.2%
44	1746	-8.62E-04	0.1%	0.0%	8.31E+07	0.5%	0.3%
45	1845	-8.61E-04	0.1%	0.0%	8.25E+07	0.7%	0.1%
46	1954	-8.61E-04	0.1%	0.0%	8.33E+07	1.0%	0.4%
47	2037	-8.61E-04	0.0%	0.0%	8.10E+07	2.7%	0.3%
48	2098	-8.62E-04	0.1%	0.0%	8.28E+07	2.1%	0.2%
49	2199	-8.61E-04	0.0%	0.0%	8.27E+07	0.1%	0.2%
50	2296	-8.62E-04	0.0%	-	8.21E+07	0.7%	0.0%

A.5 Triangular Membrane Thickness Study

The following table shows the results from the triangular membrane thickness study, comparing the M3D3 membrane and S3R shell elements. The model properties are: modulus of elasticity (303 GPa), Poison's ratio (0.37) and 1296 elements. The Newton Raphson with adaptive automatic stabilization technique is used for the FEA. t refers to the membrane thickness. U_3, S and SE refer to the center out-of-plane displacement, von Mises stress and strain energy, respectively. In case of the shell elements, stresses are given at the middle cross-sectional point. The stress % difference is based of the shell middle stress, S_{middle}, and the analytical stress, S.

Table 13: Triangular Membrane Thickness Study Results

t (m)	M3D3 Membrane Element			S3R Shell Element			% Difference		
	U_3 (m)	S (Pa)	SE (J)	U_3 (m)	S_{middle} (Pa)	SE (J)	U_3 (m)	S (Pa)	SE (J)
1.00E-06	-8.82E-03	8.53E+09	1.199	-8.83E-03	8.53E+09	1.200	0.0271	0.0567	0.1075
2.00E-06	-6.94E-03	5.31E+09	0.927	-6.94E-03	5.31E+09	0.928	0.0246	0.0535	0.0981
3.00E-06	-6.04E-03	4.03E+09	0.802	-6.04E-03	4.04E+09	0.802	0.0237	0.0524	0.0946
4.00E-06	-5.48E-03	3.32E+09	0.725	-5.48E-03	3.32E+09	0.725	0.0005	0.0009	0.0014
5.00E-06	-5.08E-03	2.86E+09	0.670	-5.08E-03	2.86E+09	0.670	0.0004	0.0007	0.0012
6.00E-06	-4.78E-03	2.53E+09	0.629	-4.78E-03	2.53E+09	0.629	0.0009	0.0016	0.0033
7.00E-06	-4.54E-03	2.28E+09	0.596	-4.54E-03	2.28E+09	0.596	0.0008	0.0014	0.0030
8.00E-06	-4.34E-03	2.08E+09	0.570	-4.34E-03	2.08E+09	0.570	0.0002	0.0003	0.0005
9.00E-06	-4.17E-03	1.93E+09	0.547	-4.17E-03	1.93E+09	0.547	0.0000	0.0001	0.0004
1.00E-05	-4.02E-03	1.79E+09	0.528	-4.02E-03	1.79E+09	0.528	0.0001	0.0000	0.0002
2.00E-05	-3.19E-03	1.13E+09	0.416	-3.19E-03	1.13E+09	0.416	0.0015	0.0005	0.0014
3.00E-05	-2.78E-03	8.60E+08	0.363	-2.78E-03	8.60E+08	0.363	0.0022	0.0022	0.0083
4.00E-05	-2.53E-03	7.09E+08	0.329	-2.53E-03	7.09E+08	0.329	0.0018	0.0063	0.0185
5.00E-05	-2.34E-03	6.11E+08	0.306	-2.34E-03	6.11E+08	0.306	0.0014	0.0128	0.0337
6.00E-05	-2.21E-03	5.41E+08	0.287	-2.21E-03	5.41E+08	0.288	0.0019	0.0209	0.0550
7.00E-05	-2.09E-03	4.88E+08	0.273	-2.09E-03	4.88E+08	0.273	0.0022	0.0323	0.0799
8.00E-05	-2.00E-03	4.46E+08	0.261	-2.00E-03	4.46E+08	0.261	0.0041	0.0447	0.1145
9.00E-05	-1.93E-03	4.13E+08	0.251	-1.93E-03	4.12E+08	0.251	0.0050	0.0615	0.1518
1.00E-04	-1.86E-03	3.85E+08	0.242	-1.86E-03	3.84E+08	0.243	0.0051	0.0827	0.1915
2.00E-04	-1.48E-03	2.42E+08	0.192	-1.48E-03	2.41E+08	0.194	0.0162	0.4738	1.0066
3.00E-04	-1.29E-03	1.85E+08	0.168	-1.29E-03	1.82E+08	0.172	0.0406	1.3581	2.4043

Continued on next page

Table 13 – continued from previous page

$t\ (m)$	M3D3 Membrane Element			S3R Shell Element			% Difference		
	$U_3\ (m)$	$S\ (Pa)$	$SE\ (J)$	$U_3\ (m)$	$S_{middle}\ (Pa)$	$SE\ (J)$	$U_3\ (m)$	$S\ (Pa)$	$SE\ (J)$
4.00E-04	-1.17E-03	1.52E+08	0.152	-1.17E-03	1.48E+08	0.159	0.2546	2.8519	4.2836
5.00E-04	-1.09E-03	1.31E+08	0.141	-1.08E-03	1.25E+08	0.151	0.7878	5.1426	6.5166
6.00E-04	-1.02E-03	1.16E+08	0.133	-1.00E-03	1.07E+08	0.145	1.9378	8.6649	8.7707
7.00E-04	-9.71E-04	1.05E+08	0.126	-9.33E-04	9.13E+07	0.141	4.0108	13.8849	10.8157
8.00E-04	-9.29E-04	9.60E+07	0.121	-8.64E-04	7.76E+07	0.137	7.2669	21.2229	12.3652
9.00E-04	-8.93E-04	8.88E+07	0.116	-7.93E-04	6.49E+07	0.132	11.9447	31.0949	12.9145
1.00E-03	-8.62E-04	8.28E+07	0.112	-7.18E-04	5.31E+07	0.126	18.2097	43.7233	11.9335
2.00E-03	-6.84E-04	5.21E+07	0.089	-1.64E-04	2.73E+06	0.035	122.8100	180.1238	86.3467

A.6 Triangular Membrane Material Properties Study

The following table shows the results of the triangular membrane material properties study for 400 M3D3 membrane elements. This study relates the changes in material properties with the icosahedral skin response, where the skin is represented by the triangle. The latter assumes that the frame remains rigid. The Newton Raphson with adaptive automatic stabilization technique is used for the FEA. The ideal W/B is set at 0.4 (see Equation (2.14)) and the skin thickness, t, results from the given density and set W/B (see Equation (2.15)). A three-dimensional input space was created with the following parameters:

$$412 \leq \rho \leq 3000 \quad kg/m^3$$
$$100 \leq E \leq 1000 \quad GPa$$
$$0.1 \leq \nu \leq 0.4$$

where:

ρ = density

E = modulus of elasticity

ν = Poison's ratio

Given the each input variable combination, the following results were considered: center out-of-plane displacement(U_3), strain energy($SE\,(J)$) and skin weight-to-buoyancy ratio (W/B_{skin}). The final skin W/B is calculated by including an estimation of the volume lost due to the triangle's deflection. This volume loss is estimated by numerical integration of the triangular surface once deflected. The same study was conducted with shell elements, but is not included here since it provided the same results.

Table 14: Triangular Membrane Material Properties Study Results

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
412	5.00E-05	1.00E+11	-3.68E-03	-3.58E-03	-3.47E-03	-3.34E-03	0.4770	0.4663	0.4529	0.4364	0.4174	0.4170	0.4165	0.4159
412	5.00E-05	1.38E+11	-3.30E-03	-3.22E-03	-3.12E-03	-3.00E-03	0.4280	0.4185	0.4065	0.3917	0.4156	0.4152	0.4148	0.4142
412	5.00E-05	1.75E+11	-3.05E-03	-2.97E-03	-2.87E-03	-2.77E-03	0.3944	0.3857	0.3746	0.3610	0.4143	0.4140	0.4136	0.4131
412	5.00E-05	2.13E+11	-2.86E-03	-2.78E-03	-2.69E-03	-2.59E-03	0.3694	0.3612	0.3508	0.3381	0.4134	0.4131	0.4127	0.4122
412	5.00E-05	2.50E+11	-2.70E-03	-2.63E-03	-2.55E-03	-2.45E-03	0.3495	0.3418	0.3320	0.3200	0.4126	0.4124	0.4120	0.4115
412	5.00E-05	2.88E+11	-2.58E-03	-2.51E-03	-2.43E-03	-2.34E-03	0.3334	0.3260	0.3167	0.3052	0.4120	0.4118	0.4114	0.4110
412	5.00E-05	3.25E+11	-2.48E-03	-2.41E-03	-2.34E-03	-2.25E-03	0.3199	0.3128	0.3039	0.2929	0.4115	0.4113	0.4110	0.4106
412	5.00E-05	3.63E+11	-2.39E-03	-2.33E-03	-2.25E-03	-2.17E-03	0.3084	0.3016	0.2929	0.2823	0.4111	0.4109	0.4106	0.4102
412	5.00E-05	4.00E+11	-2.31E-03	-2.25E-03	-2.18E-03	-2.10E-03	0.2983	0.2917	0.2834	0.2732	0.4108	0.4105	0.4102	0.4098
412	5.00E-05	4.38E+11	-2.24E-03	-2.18E-03	-2.12E-03	-2.04E-03	0.2895	0.2831	0.2750	0.2651	0.4104	0.4102	0.4099	0.4095
412	5.00E-05	4.75E+11	-2.18E-03	-2.13E-03	-2.06E-03	-1.98E-03	0.2816	0.2754	0.2675	0.2578	0.4101	0.4099	0.4096	0.4093
412	5.00E-05	5.13E+11	-2.13E-03	-2.07E-03	-2.01E-03	-1.93E-03	0.2745	0.2684	0.2608	0.2514	0.4099	0.4097	0.4094	0.4090
412	5.00E-05	5.50E+11	-2.08E-03	-2.02E-03	-1.96E-03	-1.89E-03	0.2681	0.2622	0.2547	0.2455	0.4096	0.4094	0.4091	0.4088
412	5.00E-05	5.88E+11	-2.03E-03	-1.98E-03	-1.92E-03	-1.84E-03	0.2623	0.2565	0.2492	0.2402	0.4094	0.4092	0.4089	0.4086
412	5.00E-05	6.25E+11	-1.99E-03	-1.94E-03	-1.88E-03	-1.81E-03	0.2569	0.2512	0.2440	0.2352	0.4092	0.4090	0.4088	0.4084
412	5.00E-05	6.63E+11	-1.95E-03	-1.90E-03	-1.84E-03	-1.77E-03	0.2519	0.2463	0.2393	0.2307	0.4090	0.4088	0.4086	0.4083
412	5.00E-05	7.00E+11	-1.92E-03	-1.87E-03	-1.81E-03	-1.74E-03	0.2473	0.2418	0.2349	0.2265	0.4089	0.4087	0.4084	0.4081
412	5.00E-05	7.38E+11	-1.88E-03	-1.83E-03	-1.78E-03	-1.71E-03	0.2430	0.2376	0.2309	0.2225	0.4087	0.4085	0.4083	0.4080
412	5.00E-05	7.75E+11	-1.85E-03	-1.80E-03	-1.75E-03	-1.68E-03	0.2390	0.2337	0.2271	0.2189	0.4086	0.4084	0.4081	0.4078
412	5.00E-05	8.13E+11	-1.82E-03	-1.78E-03	-1.72E-03	-1.65E-03	0.2352	0.2300	0.2235	0.2154	0.4084	0.4082	0.4080	0.4077
412	5.00E-05	8.50E+11	-1.80E-03	-1.75E-03	-1.69E-03	-1.63E-03	0.2317	0.2266	0.2201	0.2121	0.4083	0.4081	0.4079	0.4076
412	5.00E-05	8.88E+11	-1.77E-03	-1.72E-03	-1.67E-03	-1.61E-03	0.2283	0.2233	0.2169	0.2091	0.4082	0.4080	0.4078	0.4075
412	5.00E-05	9.25E+11	-1.75E-03	-1.70E-03	-1.65E-03	-1.58E-03	0.2252	0.2202	0.2139	0.2063	0.4081	0.4079	0.4077	0.4074
412	5.00E-05	9.63E+11	-1.72E-03	-1.68E-03	-1.63E-03	-1.56E-03	0.2223	0.2174	0.2112	0.2035	0.4080	0.4078	0.4076	0.4073
412	5.00E-05	1.00E+12	-1.70E-03	-1.66E-03	-1.61E-03	-1.54E-03	0.2194	0.2146	0.2085	0.2010	0.4079	0.4077	0.4075	0.4072
520	3.96E-05	1.00E+11	-3.98E-03	-3.87E-03	-3.75E-03	-3.61E-03	0.5165	0.5049	0.4903	0.4724	0.4189	0.4185	0.4179	0.4172

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
520	3.96E-05	1.38E+11	-3.57E-03	-3.48E-03	-3.37E-03	-3.24E-03	0.4633	0.4529	0.4399	0.4239	0.4169	0.4165	0.4160	0.4154
520	3.96E-05	1.75E+11	-3.30E-03	-3.21E-03	-3.11E-03	-2.99E-03	0.4268	0.4173	0.4053	0.3906	0.4155	0.4152	0.4147	0.4142
520	3.96E-05	2.13E+11	-3.09E-03	-3.01E-03	-2.91E-03	-2.80E-03	0.3996	0.3907	0.3795	0.3657	0.4145	0.4142	0.4138	0.4132
520	3.96E-05	2.50E+11	-2.92E-03	-2.85E-03	-2.76E-03	-2.65E-03	0.3782	0.3698	0.3592	0.3462	0.4137	0.4134	0.4130	0.4125
520	3.96E-05	2.88E+11	-2.79E-03	-2.72E-03	-2.63E-03	-2.53E-03	0.3607	0.3527	0.3426	0.3302	0.4131	0.4128	0.4124	0.4119
520	3.96E-05	3.25E+11	-2.68E-03	-2.61E-03	-2.53E-03	-2.43E-03	0.3461	0.3384	0.3287	0.3168	0.4125	0.4122	0.4119	0.4114
520	3.96E-05	3.63E+11	-2.58E-03	-2.51E-03	-2.44E-03	-2.34E-03	0.3335	0.3261	0.3168	0.3053	0.4121	0.4118	0.4114	0.4110
520	3.96E-05	4.00E+11	-2.50E-03	-2.43E-03	-2.36E-03	-2.27E-03	0.3227	0.3155	0.3065	0.2954	0.4116	0.4114	0.4111	0.4106
520	3.96E-05	4.38E+11	-2.42E-03	-2.36E-03	-2.29E-03	-2.20E-03	0.3131	0.3061	0.2974	0.2866	0.4113	0.4110	0.4107	0.4103
520	3.96E-05	4.75E+11	-2.36E-03	-2.30E-03	-2.22E-03	-2.14E-03	0.3045	0.2978	0.2893	0.2788	0.4110	0.4107	0.4104	0.4100
520	3.96E-05	5.13E+11	-2.30E-03	-2.24E-03	-2.17E-03	-2.09E-03	0.2968	0.2903	0.2820	0.2718	0.4107	0.4105	0.4101	0.4098
520	3.96E-05	5.50E+11	-2.25E-03	-2.19E-03	-2.12E-03	-2.04E-03	0.2899	0.2835	0.2754	0.2654	0.4104	0.4102	0.4099	0.4095
520	3.96E-05	5.88E+11	-2.20E-03	-2.14E-03	-2.07E-03	-1.99E-03	0.2835	0.2772	0.2693	0.2596	0.4102	0.4100	0.4097	0.4093
520	3.96E-05	6.25E+11	-2.15E-03	-2.10E-03	-2.03E-03	-1.95E-03	0.2777	0.2715	0.2638	0.2543	0.4100	0.4098	0.4095	0.4091
520	3.96E-05	6.63E+11	-2.11E-03	-2.06E-03	-1.99E-03	-1.91E-03	0.2723	0.2664	0.2588	0.2494	0.4098	0.4096	0.4093	0.4090
520	3.96E-05	7.00E+11	-2.07E-03	-2.02E-03	-1.95E-03	-1.88E-03	0.2674	0.2615	0.2540	0.2448	0.4096	0.4094	0.4091	0.4088
520	3.96E-05	7.38E+11	-2.04E-03	-1.98E-03	-1.92E-03	-1.85E-03	0.2627	0.2569	0.2496	0.2406	0.4094	0.4092	0.4090	0.4086
520	3.96E-05	7.75E+11	-2.00E-03	-1.95E-03	-1.89E-03	-1.82E-03	0.2584	0.2527	0.2455	0.2366	0.4093	0.4091	0.4088	0.4085
520	3.96E-05	8.13E+11	-1.97E-03	-1.92E-03	-1.86E-03	-1.79E-03	0.2543	0.2487	0.2416	0.2329	0.4091	0.4089	0.4087	0.4083
520	3.96E-05	8.50E+11	-1.94E-03	-1.89E-03	-1.83E-03	-1.76E-03	0.2505	0.2450	0.2380	0.2293	0.4090	0.4088	0.4085	0.4082
520	3.96E-05	8.88E+11	-1.91E-03	-1.86E-03	-1.81E-03	-1.74E-03	0.2469	0.2414	0.2345	0.2260	0.4089	0.4087	0.4084	0.4081
520	3.96E-05	9.25E+11	-1.89E-03	-1.84E-03	-1.78E-03	-1.71E-03	0.2434	0.2381	0.2313	0.2229	0.4087	0.4085	0.4083	0.4080
520	3.96E-05	9.63E+11	-1.86E-03	-1.81E-03	-1.76E-03	-1.69E-03	0.2402	0.2349	0.2282	0.2200	0.4086	0.4084	0.4082	0.4079
520	3.96E-05	1.00E+12	-1.84E-03	-1.79E-03	-1.73E-03	-1.67E-03	0.2372	0.2319	0.2253	0.2172	0.4085	0.4083	0.4081	0.4078
628	3.28E-05	1.00E+11	-4.24E-03	-4.13E-03	-4.00E-03	-3.84E-03	0.5510	0.5386	0.5231	0.5039	0.4202	0.4197	0.4191	0.4184
628	3.28E-05	1.38E+11	-3.81E-03	-3.71E-03	-3.59E-03	-3.45E-03	0.4941	0.4830	0.4691	0.4520	0.4180	0.4176	0.4171	0.4165
628	3.28E-05	1.75E+11	-3.51E-03	-3.42E-03	-3.31E-03	-3.18E-03	0.4551	0.4449	0.4321	0.4164	0.4166	0.4162	0.4157	0.4151
628	3.28E-05	2.13E+11	-3.29E-03	-3.20E-03	-3.10E-03	-2.98E-03	0.4260	0.4165	0.4046	0.3899	0.4155	0.4151	0.4147	0.4141

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
628	3.28E-05	2.50E+11	-3.11E-03	-3.03E-03	-2.94E-03	-2.83E-03	0.4032	0.3942	0.3829	0.3690	0.4146	0.4143	0.4139	0.4134
628	3.28E-05	2.88E+11	-2.97E-03	-2.89E-03	-2.80E-03	-2.70E-03	0.3845	0.3760	0.3652	0.3520	0.4139	0.4136	0.4132	0.4127
628	3.28E-05	3.25E+11	-2.85E-03	-2.78E-03	-2.69E-03	-2.59E-03	0.3689	0.3607	0.3504	0.3377	0.4134	0.4131	0.4127	0.4122
628	3.28E-05	3.63E+11	-2.75E-03	-2.68E-03	-2.59E-03	-2.50E-03	0.3555	0.3476	0.3376	0.3254	0.4129	0.4126	0.4122	0.4118
628	3.28E-05	4.00E+11	-2.66E-03	-2.59E-03	-2.51E-03	-2.41E-03	0.3440	0.3362	0.3267	0.3149	0.4124	0.4122	0.4118	0.4114
628	3.28E-05	4.38E+11	-2.58E-03	-2.52E-03	-2.44E-03	-2.34E-03	0.3337	0.3263	0.3169	0.3054	0.4121	0.4118	0.4114	0.4110
628	3.28E-05	4.75E+11	-2.51E-03	-2.45E-03	-2.37E-03	-2.28E-03	0.3245	0.3173	0.3082	0.2971	0.4117	0.4115	0.4111	0.4107
628	3.28E-05	5.13E+11	-2.45E-03	-2.39E-03	-2.31E-03	-2.22E-03	0.3163	0.3093	0.3005	0.2896	0.4114	0.4112	0.4108	0.4104
628	3.28E-05	5.50E+11	-2.39E-03	-2.33E-03	-2.26E-03	-2.17E-03	0.3089	0.3020	0.2934	0.2828	0.4111	0.4109	0.4106	0.4102
628	3.28E-05	5.88E+11	-2.34E-03	-2.28E-03	-2.21E-03	-2.12E-03	0.3021	0.2954	0.2870	0.2766	0.4109	0.4106	0.4103	0.4100
628	3.28E-05	6.25E+11	-2.29E-03	-2.23E-03	-2.16E-03	-2.08E-03	0.2959	0.2893	0.2811	0.2709	0.4107	0.4104	0.4101	0.4097
628	3.28E-05	6.63E+11	-2.25E-03	-2.19E-03	-2.12E-03	-2.04E-03	0.2901	0.2837	0.2756	0.2656	0.4104	0.4102	0.4099	0.4095
628	3.28E-05	7.00E+11	-2.21E-03	-2.15E-03	-2.08E-03	-2.00E-03	0.2848	0.2785	0.2706	0.2608	0.4103	0.4100	0.4097	0.4094
628	3.28E-05	7.38E+11	-2.17E-03	-2.11E-03	-2.05E-03	-1.97E-03	0.2799	0.2737	0.2659	0.2562	0.4101	0.4098	0.4096	0.4092
628	3.28E-05	7.75E+11	-2.13E-03	-2.08E-03	-2.01E-03	-1.94E-03	0.2752	0.2691	0.2615	0.2520	0.4099	0.4097	0.4094	0.4090
628	3.28E-05	8.13E+11	-2.10E-03	-2.04E-03	-1.98E-03	-1.91E-03	0.2709	0.2649	0.2574	0.2481	0.4097	0.4095	0.4092	0.4089
628	3.28E-05	8.50E+11	-2.07E-03	-2.01E-03	-1.95E-03	-1.88E-03	0.2668	0.2609	0.2535	0.2443	0.4096	0.4094	0.4091	0.4088
628	3.28E-05	8.88E+11	-2.04E-03	-1.99E-03	-1.92E-03	-1.85E-03	0.2630	0.2572	0.2498	0.2408	0.4095	0.4092	0.4090	0.4086
628	3.28E-05	9.25E+11	-2.01E-03	-1.96E-03	-1.90E-03	-1.82E-03	0.2594	0.2536	0.2464	0.2375	0.4093	0.4091	0.4088	0.4085
628	3.28E-05	9.63E+11	-1.98E-03	-1.93E-03	-1.87E-03	-1.80E-03	0.2559	0.2503	0.2431	0.2343	0.4092	0.4090	0.4087	0.4084
628	3.28E-05	1.00E+12	-1.96E-03	-1.91E-03	-1.85E-03	-1.78E-03	0.2527	0.2471	0.2400	0.2314	0.4091	0.4089	0.4086	0.4083
736	2.80E-05	1.00E+11	-4.47E-03	-4.35E-03	-4.21E-03	-4.05E-03	0.5820	0.5688	0.5523	0.5321	0.4214	0.4209	0.4202	0.4195
736	2.80E-05	1.38E+11	-4.01E-03	-3.91E-03	-3.79E-03	-3.64E-03	0.5217	0.5100	0.4952	0.4771	0.4191	0.4186	0.4181	0.4174
736	2.80E-05	1.75E+11	-3.70E-03	-3.61E-03	-3.49E-03	-3.36E-03	0.4804	0.4696	0.4561	0.4395	0.4175	0.4171	0.4166	0.4160
736	2.80E-05	2.13E+11	-3.47E-03	-3.38E-03	-3.27E-03	-3.15E-03	0.4496	0.4396	0.4269	0.4114	0.4164	0.4160	0.4155	0.4149
736	2.80E-05	2.50E+11	-3.28E-03	-3.20E-03	-3.10E-03	-2.98E-03	0.4255	0.4160	0.4040	0.3894	0.4155	0.4151	0.4147	0.4141
736	2.80E-05	2.88E+11	-3.13E-03	-3.05E-03	-2.96E-03	-2.84E-03	0.4058	0.3967	0.3854	0.3714	0.4147	0.4144	0.4140	0.4135
736	2.80E-05	3.25E+11	-3.01E-03	-2.93E-03	-2.84E-03	-2.73E-03	0.3893	0.3806	0.3697	0.3563	0.4141	0.4138	0.4134	0.4129

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
736	2.80E-05	3.63E+11	-2.90E-03	-2.82E-03	-2.74E-03	-2.63E-03	0.3752	0.3668	0.3563	0.3434	0.4136	0.4133	0.4129	0.4124
736	2.80E-05	4.00E+11	-2.81E-03	-2.73E-03	-2.65E-03	-2.55E-03	0.3629	0.3547	0.3447	0.3322	0.4131	0.4128	0.4125	0.4120
736	2.80E-05	4.38E+11	-2.72E-03	-2.65E-03	-2.57E-03	-2.47E-03	0.3521	0.3443	0.3343	0.3222	0.4127	0.4124	0.4121	0.4116
736	2.80E-05	4.75E+11	-2.65E-03	-2.58E-03	-2.50E-03	-2.40E-03	0.3423	0.3347	0.3252	0.3134	0.4124	0.4121	0.4117	0.4113
736	2.80E-05	5.13E+11	-2.58E-03	-2.52E-03	-2.44E-03	-2.34E-03	0.3337	0.3263	0.3169	0.3055	0.4121	0.4118	0.4114	0.4110
736	2.80E-05	5.50E+11	-2.52E-03	-2.46E-03	-2.38E-03	-2.29E-03	0.3258	0.3186	0.3095	0.2983	0.4118	0.4115	0.4112	0.4107
736	2.80E-05	5.88E+11	-2.47E-03	-2.40E-03	-2.33E-03	-2.24E-03	0.3187	0.3116	0.3027	0.2917	0.4115	0.4112	0.4109	0.4105
736	2.80E-05	6.25E+11	-2.42E-03	-2.35E-03	-2.28E-03	-2.19E-03	0.3121	0.3052	0.2965	0.2857	0.4113	0.4110	0.4107	0.4103
736	2.80E-05	6.63E+11	-2.37E-03	-2.31E-03	-2.24E-03	-2.15E-03	0.3060	0.2993	0.2907	0.2802	0.4110	0.4108	0.4105	0.4101
736	2.80E-05	7.00E+11	-2.33E-03	-2.27E-03	-2.19E-03	-2.11E-03	0.3004	0.2938	0.2854	0.2751	0.4108	0.4106	0.4103	0.4099
736	2.80E-05	7.38E+11	-2.29E-03	-2.23E-03	-2.16E-03	-2.07E-03	0.2952	0.2887	0.2804	0.2703	0.4106	0.4104	0.4101	0.4097
736	2.80E-05	7.75E+11	-2.25E-03	-2.19E-03	-2.12E-03	-2.04E-03	0.2903	0.2839	0.2758	0.2658	0.4105	0.4102	0.4099	0.4096
736	2.80E-05	8.13E+11	-2.21E-03	-2.16E-03	-2.09E-03	-2.01E-03	0.2857	0.2794	0.2714	0.2616	0.4103	0.4101	0.4098	0.4094
736	2.80E-05	8.50E+11	-2.18E-03	-2.12E-03	-2.06E-03	-1.98E-03	0.2814	0.2752	0.2674	0.2577	0.4101	0.4099	0.4096	0.4093
736	2.80E-05	8.88E+11	-2.15E-03	-2.09E-03	-2.03E-03	-1.95E-03	0.2774	0.2712	0.2635	0.2540	0.4100	0.4098	0.4095	0.4091
736	2.80E-05	9.25E+11	-2.12E-03	-2.06E-03	-2.00E-03	-1.92E-03	0.2736	0.2675	0.2599	0.2505	0.4098	0.4096	0.4093	0.4090
736	2.80E-05	9.63E+11	-2.09E-03	-2.04E-03	-1.97E-03	-1.90E-03	0.2699	0.2640	0.2564	0.2472	0.4097	0.4095	0.4092	0.4089
736	2.80E-05	1.00E+12	-2.07E-03	-2.01E-03	-1.95E-03	-1.87E-03	0.2665	0.2606	0.2532	0.2440	0.4096	0.4094	0.4091	0.4088
843	2.44E-05	1.00E+11	-4.68E-03	-4.55E-03	-4.41E-03	-4.24E-03	0.6099	0.5961	0.5788	0.5575	0.4224	0.4219	0.4213	0.4204
843	2.44E-05	1.38E+11	-4.20E-03	-4.09E-03	-3.96E-03	-3.81E-03	0.5465	0.5342	0.5188	0.4998	0.4200	0.4196	0.4190	0.4183
843	2.44E-05	1.75E+11	-3.87E-03	-3.77E-03	-3.65E-03	-3.52E-03	0.5031	0.4919	0.4777	0.4603	0.4184	0.4180	0.4174	0.4168
843	2.44E-05	2.13E+11	-3.63E-03	-3.54E-03	-3.42E-03	-3.29E-03	0.4709	0.4604	0.4471	0.4308	0.4172	0.4168	0.4163	0.4157
843	2.44E-05	2.50E+11	-3.44E-03	-3.35E-03	-3.24E-03	-3.12E-03	0.4455	0.4356	0.4231	0.4077	0.4162	0.4158	0.4154	0.4148
843	2.44E-05	2.88E+11	-3.28E-03	-3.19E-03	-3.09E-03	-2.98E-03	0.4249	0.4154	0.4035	0.3888	0.4154	0.4151	0.4146	0.4141
843	2.44E-05	3.25E+11	-3.15E-03	-3.07E-03	-2.97E-03	-2.86E-03	0.4076	0.3985	0.3871	0.3730	0.4148	0.4145	0.4140	0.4135
843	2.44E-05	3.63E+11	-3.04E-03	-2.96E-03	-2.86E-03	-2.75E-03	0.3928	0.3840	0.3730	0.3595	0.4142	0.4139	0.4135	0.4130
843	2.44E-05	4.00E+11	-2.94E-03	-2.86E-03	-2.77E-03	-2.66E-03	0.3799	0.3715	0.3608	0.3477	0.4138	0.4135	0.4131	0.4126
843	2.44E-05	4.38E+11	-2.85E-03	-2.78E-03	-2.69E-03	-2.59E-03	0.3686	0.3604	0.3500	0.3373	0.4133	0.4130	0.4127	0.4122

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
843	2.44E-05	4.75E+11	-2.77E-03	-2.70E-03	-2.61E-03	-2.52E-03	0.3584	0.3504	0.3404	0.3281	0.4130	0.4127	0.4123	0.4118
843	2.44E-05	5.13E+11	-2.70E-03	-2.63E-03	-2.55E-03	-2.45E-03	0.3493	0.3415	0.3318	0.3198	0.4126	0.4123	0.4120	0.4115
843	2.44E-05	5.50E+11	-2.64E-03	-2.57E-03	-2.49E-03	-2.39E-03	0.3411	0.3335	0.3240	0.3122	0.4123	0.4121	0.4117	0.4113
843	2.44E-05	5.88E+11	-2.58E-03	-2.51E-03	-2.44E-03	-2.34E-03	0.3336	0.3262	0.3169	0.3054	0.4121	0.4118	0.4114	0.4110
843	2.44E-05	6.25E+11	-2.53E-03	-2.46E-03	-2.39E-03	-2.29E-03	0.3267	0.3194	0.3103	0.2991	0.4118	0.4115	0.4112	0.4108
843	2.44E-05	6.63E+11	-2.48E-03	-2.42E-03	-2.34E-03	-2.25E-03	0.3203	0.3132	0.3043	0.2933	0.4116	0.4113	0.4110	0.4106
843	2.44E-05	7.00E+11	-2.44E-03	-2.37E-03	-2.30E-03	-2.21E-03	0.3145	0.3075	0.2987	0.2879	0.4113	0.4111	0.4108	0.4104
843	2.44E-05	7.38E+11	-2.39E-03	-2.33E-03	-2.26E-03	-2.17E-03	0.3090	0.3021	0.2935	0.2829	0.4111	0.4109	0.4106	0.4102
843	2.44E-05	7.75E+11	-2.35E-03	-2.29E-03	-2.22E-03	-2.14E-03	0.3039	0.2971	0.2887	0.2782	0.4110	0.4107	0.4104	0.4100
843	2.44E-05	8.13E+11	-2.32E-03	-2.26E-03	-2.19E-03	-2.10E-03	0.2991	0.2925	0.2841	0.2738	0.4108	0.4105	0.4102	0.4098
843	2.44E-05	8.50E+11	-2.28E-03	-2.22E-03	-2.15E-03	-2.07E-03	0.2946	0.2880	0.2798	0.2697	0.4106	0.4104	0.4101	0.4097
843	2.44E-05	8.88E+11	-2.25E-03	-2.19E-03	-2.12E-03	-2.04E-03	0.2903	0.2839	0.2758	0.2658	0.4105	0.4102	0.4099	0.4096
843	2.44E-05	9.25E+11	-2.22E-03	-2.16E-03	-2.09E-03	-2.01E-03	0.2863	0.2800	0.2720	0.2622	0.4103	0.4101	0.4098	0.4094
843	2.44E-05	9.63E+11	-2.19E-03	-2.13E-03	-2.06E-03	-1.99E-03	0.2825	0.2763	0.2684	0.2587	0.4102	0.4099	0.4096	0.4093
843	2.44E-05	1.00E+12	-2.16E-03	-2.10E-03	-2.04E-03	-1.96E-03	0.2789	0.2727	0.2650	0.2554	0.4100	0.4098	0.4095	0.4092
951	2.17E-05	1.00E+11	-4.87E-03	-4.74E-03	-4.59E-03	-4.42E-03	0.6359	0.6215	0.6034	0.5812	0.4234	0.4229	0.4222	0.4213
951	2.17E-05	1.38E+11	-4.38E-03	-4.26E-03	-4.13E-03	-3.97E-03	0.5696	0.5568	0.5407	0.5209	0.4209	0.4204	0.4198	0.4191
951	2.17E-05	1.75E+11	-4.04E-03	-3.93E-03	-3.81E-03	-3.66E-03	0.5244	0.5126	0.4978	0.4796	0.4192	0.4187	0.4182	0.4175
951	2.17E-05	2.13E+11	-3.78E-03	-3.68E-03	-3.57E-03	-3.43E-03	0.4907	0.4796	0.4659	0.4489	0.4179	0.4175	0.4170	0.4163
951	2.17E-05	2.50E+11	-3.58E-03	-3.49E-03	-3.38E-03	-3.25E-03	0.4642	0.4538	0.4408	0.4247	0.4169	0.4165	0.4160	0.4154
951	2.17E-05	2.88E+11	-3.42E-03	-3.33E-03	-3.22E-03	-3.10E-03	0.4426	0.4328	0.4203	0.4050	0.4161	0.4157	0.4153	0.4147
951	2.17E-05	3.25E+11	-3.28E-03	-3.19E-03	-3.09E-03	-2.97E-03	0.4246	0.4151	0.4032	0.3886	0.4154	0.4151	0.4146	0.4141
951	2.17E-05	3.63E+11	-3.16E-03	-3.08E-03	-2.98E-03	-2.87E-03	0.4092	0.4000	0.3886	0.3744	0.4149	0.4145	0.4141	0.4136
951	2.17E-05	4.00E+11	-3.06E-03	-2.98E-03	-2.88E-03	-2.77E-03	0.3957	0.3869	0.3758	0.3622	0.4144	0.4140	0.4136	0.4131
951	2.17E-05	4.38E+11	-2.97E-03	-2.89E-03	-2.80E-03	-2.69E-03	0.3839	0.3754	0.3645	0.3513	0.4139	0.4136	0.4132	0.4127
951	2.17E-05	4.75E+11	-2.89E-03	-2.81E-03	-2.72E-03	-2.62E-03	0.3733	0.3650	0.3545	0.3417	0.4135	0.4132	0.4128	0.4124
951	2.17E-05	5.13E+11	-2.81E-03	-2.74E-03	-2.65E-03	-2.55E-03	0.3638	0.3557	0.3455	0.3330	0.4132	0.4129	0.4125	0.4120
951	2.17E-05	5.50E+11	-2.75E-03	-2.68E-03	-2.59E-03	-2.49E-03	0.3552	0.3473	0.3374	0.3252	0.4129	0.4126	0.4122	0.4117

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
951	2.17E-05	5.88E+11	-2.69E-03	-2.62E-03	-2.54E-03	-2.44E-03	0.3474	0.3397	0.3300	0.3180	0.4126	0.4123	0.4119	0.4115
951	2.17E-05	6.25E+11	-2.63E-03	-2.56E-03	-2.48E-03	-2.39E-03	0.3402	0.3327	0.3232	0.3115	0.4123	0.4120	0.4117	0.4112
951	2.17E-05	6.63E+11	-2.58E-03	-2.51E-03	-2.44E-03	-2.34E-03	0.3336	0.3262	0.3169	0.3054	0.4121	0.4118	0.4114	0.4110
951	2.17E-05	7.00E+11	-2.54E-03	-2.47E-03	-2.39E-03	-2.30E-03	0.3275	0.3202	0.3111	0.2998	0.4118	0.4116	0.4112	0.4108
951	2.17E-05	7.38E+11	-2.49E-03	-2.43E-03	-2.35E-03	-2.26E-03	0.3218	0.3146	0.3057	0.2946	0.4116	0.4114	0.4110	0.4106
951	2.17E-05	7.75E+11	-2.45E-03	-2.39E-03	-2.31E-03	-2.22E-03	0.3164	0.3094	0.3006	0.2897	0.4114	0.4112	0.4108	0.4104
951	2.17E-05	8.13E+11	-2.41E-03	-2.35E-03	-2.27E-03	-2.19E-03	0.3115	0.3046	0.2959	0.2852	0.4112	0.4110	0.4107	0.4103
951	2.17E-05	8.50E+11	-2.38E-03	-2.31E-03	-2.24E-03	-2.16E-03	0.3068	0.3000	0.2914	0.2809	0.4111	0.4108	0.4105	0.4101
951	2.17E-05	8.88E+11	-2.34E-03	-2.28E-03	-2.21E-03	-2.12E-03	0.3023	0.2956	0.2872	0.2768	0.4109	0.4107	0.4103	0.4100
951	2.17E-05	9.25E+11	-2.31E-03	-2.25E-03	-2.18E-03	-2.10E-03	0.2982	0.2916	0.2832	0.2730	0.4107	0.4105	0.4102	0.4098
951	2.17E-05	9.63E+11	-2.28E-03	-2.22E-03	-2.15E-03	-2.07E-03	0.2942	0.2877	0.2795	0.2694	0.4106	0.4104	0.4101	0.4097
951	2.17E-05	1.00E+12	-2.25E-03	-2.19E-03	-2.12E-03	-2.04E-03	0.2904	0.2840	0.2759	0.2659	0.4105	0.4102	0.4099	0.4096
1059	1.94E-05	1.00E+11	-5.05E-03	-4.92E-03	-4.76E-03	-4.58E-03	0.6601	0.6451	0.6263	0.6032	0.4243	0.4238	0.4231	0.4222
1059	1.94E-05	1.38E+11	-4.54E-03	-4.42E-03	-4.28E-03	-4.12E-03	0.5911	0.5778	0.5610	0.5404	0.4217	0.4212	0.4206	0.4198
1059	1.94E-05	1.75E+11	-4.18E-03	-4.07E-03	-3.95E-03	-3.80E-03	0.5441	0.5318	0.5165	0.4976	0.4199	0.4195	0.4189	0.4182
1059	1.94E-05	2.13E+11	-3.92E-03	-3.82E-03	-3.70E-03	-3.56E-03	0.5091	0.4976	0.4833	0.4656	0.4186	0.4182	0.4176	0.4170
1059	1.94E-05	2.50E+11	-3.71E-03	-3.61E-03	-3.50E-03	-3.37E-03	0.4815	0.4708	0.4572	0.4405	0.4176	0.4172	0.4167	0.4160
1059	1.94E-05	2.88E+11	-3.54E-03	-3.45E-03	-3.34E-03	-3.21E-03	0.4592	0.4489	0.4360	0.4201	0.4167	0.4163	0.4159	0.4153
1059	1.94E-05	3.25E+11	-3.40E-03	-3.31E-03	-3.21E-03	-3.08E-03	0.4404	0.4306	0.4182	0.4030	0.4160	0.4157	0.4152	0.4146
1059	1.94E-05	3.63E+11	-3.28E-03	-3.19E-03	-3.09E-03	-2.97E-03	0.4244	0.4149	0.4030	0.3883	0.4154	0.4151	0.4146	0.4141
1059	1.94E-05	4.00E+11	-3.17E-03	-3.09E-03	-2.99E-03	-2.88E-03	0.4104	0.4013	0.3898	0.3756	0.4149	0.4146	0.4141	0.4136
1059	1.94E-05	4.38E+11	-3.08E-03	-3.00E-03	-2.90E-03	-2.79E-03	0.3982	0.3893	0.3780	0.3643	0.4144	0.4141	0.4137	0.4132
1059	1.94E-05	4.75E+11	-2.99E-03	-2.91E-03	-2.82E-03	-2.72E-03	0.3871	0.3785	0.3676	0.3543	0.4140	0.4137	0.4133	0.4128
1059	1.94E-05	5.13E+11	-2.92E-03	-2.84E-03	-2.75E-03	-2.65E-03	0.3773	0.3689	0.3583	0.3453	0.4137	0.4134	0.4130	0.4125
1059	1.94E-05	5.50E+11	-2.85E-03	-2.77E-03	-2.69E-03	-2.58E-03	0.3684	0.3602	0.3499	0.3372	0.4133	0.4130	0.4127	0.4122
1059	1.94E-05	5.88E+11	-2.79E-03	-2.71E-03	-2.63E-03	-2.53E-03	0.3603	0.3523	0.3422	0.3298	0.4130	0.4127	0.4124	0.4119
1059	1.94E-05	6.25E+11	-2.73E-03	-2.66E-03	-2.57E-03	-2.48E-03	0.3528	0.3450	0.3351	0.3230	0.4128	0.4125	0.4121	0.4117
1059	1.94E-05	6.63E+11	-2.68E-03	-2.61E-03	-2.52E-03	-2.43E-03	0.3459	0.3383	0.3286	0.3167	0.4125	0.4122	0.4119	0.4114

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1059	1.94E-05	7.00E+11	-2.63E-03	-2.56E-03	-2.48E-03	-2.38E-03	0.3396	0.3320	0.3226	0.3109	0.4123	0.4120	0.4116	0.4112
1059	1.94E-05	7.38E+11	-2.58E-03	-2.52E-03	-2.44E-03	-2.34E-03	0.3337	0.3263	0.3169	0.3055	0.4121	0.4118	0.4114	0.4110
1059	1.94E-05	7.75E+11	-2.54E-03	-2.47E-03	-2.40E-03	-2.30E-03	0.3281	0.3208	0.3117	0.3004	0.4119	0.4116	0.4112	0.4108
1059	1.94E-05	8.13E+11	-2.50E-03	-2.43E-03	-2.36E-03	-2.27E-03	0.3229	0.3158	0.3068	0.2957	0.4117	0.4114	0.4111	0.4107
1059	1.94E-05	8.50E+11	-2.46E-03	-2.40E-03	-2.32E-03	-2.23E-03	0.3181	0.3110	0.3021	0.2912	0.4115	0.4112	0.4109	0.4105
1059	1.94E-05	8.88E+11	-2.43E-03	-2.36E-03	-2.29E-03	-2.20E-03	0.3135	0.3065	0.2978	0.2870	0.4113	0.4111	0.4107	0.4103
1059	1.94E-05	9.25E+11	-2.39E-03	-2.33E-03	-2.26E-03	-2.17E-03	0.3091	0.3023	0.2937	0.2830	0.4111	0.4109	0.4106	0.4102
1059	1.94E-05	9.63E+11	-2.36E-03	-2.30E-03	-2.23E-03	-2.14E-03	0.3050	0.2983	0.2898	0.2793	0.4110	0.4107	0.4104	0.4100
1059	1.94E-05	1.00E+12	-2.33E-03	-2.27E-03	-2.20E-03	-2.12E-03	0.3011	0.2945	0.2861	0.2757	0.4109	0.4106	0.4103	0.4099
1167	1.76E-05	1.00E+11	-5.22E-03	-5.08E-03	-4.92E-03	-4.74E-03	0.6827	0.6672	0.6477	0.6238	0.4252	0.4246	0.4239	0.4230
1167	1.76E-05	1.38E+11	-4.69E-03	-4.57E-03	-4.42E-03	-4.25E-03	0.6113	0.5975	0.5801	0.5588	0.4225	0.4220	0.4213	0.4205
1167	1.76E-05	1.75E+11	-4.32E-03	-4.21E-03	-4.08E-03	-3.92E-03	0.5625	0.5499	0.5339	0.5144	0.4206	0.4202	0.4196	0.4188
1167	1.76E-05	2.13E+11	-4.05E-03	-3.94E-03	-3.82E-03	-3.67E-03	0.5263	0.5144	0.4996	0.4813	0.4193	0.4188	0.4183	0.4176
1167	1.76E-05	2.50E+11	-3.83E-03	-3.73E-03	-3.62E-03	-3.48E-03	0.4978	0.4866	0.4726	0.4553	0.4182	0.4178	0.4172	0.4166
1167	1.76E-05	2.88E+11	-3.66E-03	-3.56E-03	-3.45E-03	-3.32E-03	0.4746	0.4640	0.4506	0.4342	0.4173	0.4169	0.4164	0.4158
1167	1.76E-05	3.25E+11	-3.51E-03	-3.42E-03	-3.31E-03	-3.19E-03	0.4552	0.4450	0.4322	0.4165	0.4166	0.4162	0.4157	0.4151
1167	1.76E-05	3.63E+11	-3.38E-03	-3.30E-03	-3.19E-03	-3.07E-03	0.4386	0.4288	0.4165	0.4013	0.4160	0.4156	0.4151	0.4146
1167	1.76E-05	4.00E+11	-3.27E-03	-3.19E-03	-3.09E-03	-2.97E-03	0.4242	0.4147	0.4028	0.3882	0.4154	0.4151	0.4146	0.4141
1167	1.76E-05	4.38E+11	-3.18E-03	-3.09E-03	-3.00E-03	-2.88E-03	0.4115	0.4023	0.3908	0.3766	0.4149	0.4146	0.4142	0.4136
1167	1.76E-05	4.75E+11	-3.09E-03	-3.01E-03	-2.92E-03	-2.80E-03	0.4000	0.3913	0.3799	0.3661	0.4145	0.4142	0.4138	0.4133
1167	1.76E-05	5.13E+11	-3.01E-03	-2.93E-03	-2.84E-03	-2.73E-03	0.3899	0.3812	0.3703	0.3568	0.4141	0.4138	0.4134	0.4129
1167	1.76E-05	5.50E+11	-2.94E-03	-2.87E-03	-2.78E-03	-2.67E-03	0.3807	0.3722	0.3615	0.3484	0.4138	0.4135	0.4131	0.4126
1167	1.76E-05	5.88E+11	-2.88E-03	-2.80E-03	-2.72E-03	-2.61E-03	0.3723	0.3640	0.3536	0.3408	0.4135	0.4132	0.4128	0.4123
1167	1.76E-05	6.25E+11	-2.82E-03	-2.75E-03	-2.66E-03	-2.56E-03	0.3646	0.3565	0.3463	0.3337	0.4132	0.4129	0.4125	0.4121
1167	1.76E-05	6.63E+11	-2.77E-03	-2.69E-03	-2.61E-03	-2.51E-03	0.3575	0.3495	0.3395	0.3272	0.4129	0.4126	0.4123	0.4118
1167	1.76E-05	7.00E+11	-2.72E-03	-2.64E-03	-2.56E-03	-2.46E-03	0.3509	0.3431	0.3333	0.3212	0.4127	0.4124	0.4120	0.4116
1167	1.76E-05	7.38E+11	-2.67E-03	-2.60E-03	-2.52E-03	-2.42E-03	0.3448	0.3371	0.3275	0.3156	0.4125	0.4122	0.4118	0.4114
1167	1.76E-05	7.75E+11	-2.62E-03	-2.56E-03	-2.47E-03	-2.38E-03	0.3390	0.3315	0.3220	0.3104	0.4123	0.4120	0.4116	0.4112

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1167	1.76E-05	8.13E+11	-2.58E-03	-2.52E-03	-2.44E-03	-2.34E-03	0.3337	0.3263	0.3170	0.3055	0.4121	0.4118	0.4114	0.4110
1167	1.76E-05	8.50E+11	-2.54E-03	-2.48E-03	-2.40E-03	-2.31E-03	0.3286	0.3214	0.3122	0.3009	0.4119	0.4116	0.4113	0.4108
1167	1.76E-05	8.88E+11	-2.51E-03	-2.44E-03	-2.37E-03	-2.28E-03	0.3239	0.3167	0.3077	0.2965	0.4117	0.4114	0.4111	0.4107
1167	1.76E-05	9.25E+11	-2.47E-03	-2.41E-03	-2.33E-03	-2.24E-03	0.3194	0.3123	0.3034	0.2924	0.4115	0.4113	0.4109	0.4105
1167	1.76E-05	9.63E+11	-2.44E-03	-2.38E-03	-2.30E-03	-2.21E-03	0.3152	0.3082	0.2994	0.2886	0.4114	0.4111	0.4108	0.4104
1167	1.76E-05	1.00E+12	-2.41E-03	-2.35E-03	-2.27E-03	-2.19E-03	0.3111	0.3043	0.2956	0.2849	0.4112	0.4110	0.4106	0.4103
1275	1.62E-05	1.00E+11	-5.38E-03	-5.24E-03	-5.07E-03	-4.88E-03	0.7042	0.6881	0.6680	0.6433	0.4260	0.4254	0.4247	0.4237
1275	1.62E-05	1.38E+11	-4.83E-03	-4.70E-03	-4.56E-03	-4.38E-03	0.6303	0.6161	0.5981	0.5761	0.4232	0.4227	0.4220	0.4212
1275	1.62E-05	1.75E+11	-4.45E-03	-4.34E-03	-4.20E-03	-4.04E-03	0.5799	0.5669	0.5504	0.5302	0.4213	0.4208	0.4202	0.4194
1275	1.62E-05	2.13E+11	-4.17E-03	-4.06E-03	-3.93E-03	-3.79E-03	0.5425	0.5303	0.5149	0.4961	0.4199	0.4194	0.4188	0.4181
1275	1.62E-05	2.50E+11	-3.95E-03	-3.85E-03	-3.73E-03	-3.58E-03	0.5131	0.5016	0.4871	0.4693	0.4188	0.4183	0.4178	0.4171
1275	1.62E-05	2.88E+11	-3.77E-03	-3.67E-03	-3.55E-03	-3.42E-03	0.4892	0.4781	0.4644	0.4475	0.4179	0.4174	0.4169	0.4163
1275	1.62E-05	3.25E+11	-3.62E-03	-3.52E-03	-3.41E-03	-3.28E-03	0.4691	0.4586	0.4454	0.4292	0.4171	0.4167	0.4162	0.4156
1275	1.62E-05	3.63E+11	-3.49E-03	-3.40E-03	-3.29E-03	-3.16E-03	0.4520	0.4419	0.4292	0.4136	0.4165	0.4161	0.4156	0.4150
1275	1.62E-05	4.00E+11	-3.37E-03	-3.29E-03	-3.18E-03	-3.06E-03	0.4370	0.4274	0.4150	0.4000	0.4159	0.4155	0.4151	0.4145
1275	1.62E-05	4.38E+11	-3.27E-03	-3.19E-03	-3.09E-03	-2.97E-03	0.4240	0.4145	0.4027	0.3880	0.4154	0.4151	0.4146	0.4141
1275	1.62E-05	4.75E+11	-3.18E-03	-3.10E-03	-3.00E-03	-2.89E-03	0.4124	0.4032	0.3916	0.3774	0.4150	0.4146	0.4142	0.4137
1275	1.62E-05	5.13E+11	-3.10E-03	-3.02E-03	-2.93E-03	-2.82E-03	0.4019	0.3929	0.3817	0.3677	0.4146	0.4143	0.4138	0.4133
1275	1.62E-05	5.50E+11	-3.03E-03	-2.95E-03	-2.86E-03	-2.75E-03	0.3922	0.3835	0.3725	0.3590	0.4142	0.4139	0.4135	0.4130
1275	1.62E-05	5.88E+11	-2.97E-03	-2.89E-03	-2.80E-03	-2.69E-03	0.3836	0.3751	0.3643	0.3511	0.4139	0.4136	0.4132	0.4127
1275	1.62E-05	6.25E+11	-2.90E-03	-2.83E-03	-2.74E-03	-2.64E-03	0.3756	0.3673	0.3568	0.3438	0.4136	0.4133	0.4129	0.4124
1275	1.62E-05	6.63E+11	-2.85E-03	-2.77E-03	-2.69E-03	-2.58E-03	0.3683	0.3601	0.3498	0.3371	0.4133	0.4130	0.4127	0.4122
1275	1.62E-05	7.00E+11	-2.80E-03	-2.72E-03	-2.64E-03	-2.54E-03	0.3615	0.3535	0.3434	0.3309	0.4131	0.4128	0.4124	0.4120
1275	1.62E-05	7.38E+11	-2.75E-03	-2.68E-03	-2.59E-03	-2.49E-03	0.3552	0.3473	0.3374	0.3252	0.4129	0.4126	0.4122	0.4117
1275	1.62E-05	7.75E+11	-2.70E-03	-2.63E-03	-2.55E-03	-2.45E-03	0.3493	0.3416	0.3318	0.3198	0.4126	0.4123	0.4120	0.4115
1275	1.62E-05	8.13E+11	-2.66E-03	-2.59E-03	-2.51E-03	-2.41E-03	0.3438	0.3362	0.3266	0.3147	0.4124	0.4121	0.4118	0.4114
1275	1.62E-05	8.50E+11	-2.62E-03	-2.55E-03	-2.47E-03	-2.38E-03	0.3386	0.3311	0.3216	0.3100	0.4122	0.4120	0.4116	0.4112
1275	1.62E-05	8.88E+11	-2.58E-03	-2.52E-03	-2.44E-03	-2.34E-03	0.3337	0.3263	0.3170	0.3055	0.4121	0.4118	0.4114	0.4110

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1275	1.62E-05	9.25E+11	-2.55E-03	-2.48E-03	-2.40E-03	-2.31E-03	0.3291	0.3218	0.3126	0.3013	0.4119	0.4116	0.4113	0.4109
1275	1.62E-05	9.63E+11	-2.51E-03	-2.45E-03	-2.37E-03	-2.28E-03	0.3247	0.3175	0.3084	0.2973	0.4117	0.4115	0.4111	0.4107
1275	1.62E-05	1.00E+12	-2.48E-03	-2.42E-03	-2.34E-03	-2.25E-03	0.3206	0.3135	0.3045	0.2935	0.4116	0.4113	0.4110	0.4106
1383	1.49E-05	1.00E+11	-5.53E-03	-5.38E-03	-5.21E-03	-5.02E-03	0.7245	0.7080	0.6872	0.6618	0.4268	0.4262	0.4254	0.4244
1383	1.49E-05	1.38E+11	-4.96E-03	-4.83E-03	-4.68E-03	-4.50E-03	0.6484	0.6337	0.6152	0.5925	0.4239	0.4233	0.4226	0.4218
1383	1.49E-05	1.75E+11	-4.58E-03	-4.46E-03	-4.32E-03	-4.15E-03	0.5964	0.5830	0.5660	0.5453	0.4219	0.4214	0.4208	0.4200
1383	1.49E-05	2.13E+11	-4.29E-03	-4.17E-03	-4.04E-03	-3.89E-03	0.5578	0.5453	0.5295	0.5101	0.4205	0.4200	0.4194	0.4187
1383	1.49E-05	2.50E+11	-4.06E-03	-3.95E-03	-3.83E-03	-3.68E-03	0.5276	0.5157	0.5008	0.4825	0.4193	0.4189	0.4183	0.4176
1383	1.49E-05	2.88E+11	-3.87E-03	-3.77E-03	-3.65E-03	-3.51E-03	0.5029	0.4917	0.4775	0.4600	0.4184	0.4180	0.4174	0.4168
1383	1.49E-05	3.25E+11	-3.72E-03	-3.62E-03	-3.51E-03	-3.37E-03	0.4823	0.4714	0.4579	0.4412	0.4176	0.4172	0.4167	0.4161
1383	1.49E-05	3.63E+11	-3.58E-03	-3.49E-03	-3.38E-03	-3.25E-03	0.4647	0.4543	0.4412	0.4251	0.4169	0.4166	0.4161	0.4155
1383	1.49E-05	4.00E+11	-3.47E-03	-3.38E-03	-3.27E-03	-3.15E-03	0.4494	0.4393	0.4266	0.4111	0.4164	0.4160	0.4155	0.4149
1383	1.49E-05	4.38E+11	-3.36E-03	-3.28E-03	-3.17E-03	-3.05E-03	0.4358	0.4261	0.4139	0.3989	0.4159	0.4155	0.4150	0.4145
1383	1.49E-05	4.75E+11	-3.27E-03	-3.19E-03	-3.09E-03	-2.97E-03	0.4239	0.4144	0.4025	0.3879	0.4154	0.4151	0.4146	0.4141
1383	1.49E-05	5.13E+11	-3.19E-03	-3.11E-03	-3.01E-03	-2.89E-03	0.4131	0.4039	0.3923	0.3781	0.4150	0.4147	0.4142	0.4137
1383	1.49E-05	5.50E+11	-3.12E-03	-3.03E-03	-2.94E-03	-2.83E-03	0.4033	0.3944	0.3830	0.3690	0.4146	0.4143	0.4139	0.4134
1383	1.49E-05	5.88E+11	-3.05E-03	-2.97E-03	-2.87E-03	-2.76E-03	0.3943	0.3855	0.3745	0.3609	0.4143	0.4140	0.4136	0.4131
1383	1.49E-05	6.25E+11	-2.99E-03	-2.91E-03	-2.82E-03	-2.71E-03	0.3861	0.3775	0.3667	0.3534	0.4140	0.4137	0.4133	0.4128
1383	1.49E-05	6.63E+11	-2.93E-03	-2.85E-03	-2.76E-03	-2.66E-03	0.3786	0.3702	0.3596	0.3465	0.4137	0.4134	0.4130	0.4125
1383	1.49E-05	7.00E+11	-2.87E-03	-2.80E-03	-2.71E-03	-2.61E-03	0.3716	0.3633	0.3529	0.3401	0.4135	0.4132	0.4128	0.4123
1383	1.49E-05	7.38E+11	-2.82E-03	-2.75E-03	-2.66E-03	-2.56E-03	0.3651	0.3570	0.3468	0.3342	0.4132	0.4129	0.4125	0.4121
1383	1.49E-05	7.75E+11	-2.78E-03	-2.70E-03	-2.62E-03	-2.52E-03	0.3590	0.3511	0.3410	0.3287	0.4130	0.4127	0.4123	0.4119
1383	1.49E-05	8.13E+11	-2.73E-03	-2.66E-03	-2.58E-03	-2.48E-03	0.3534	0.3455	0.3356	0.3235	0.4128	0.4125	0.4121	0.4117
1383	1.49E-05	8.50E+11	-2.69E-03	-2.62E-03	-2.54E-03	-2.44E-03	0.3480	0.3403	0.3306	0.3186	0.4126	0.4123	0.4119	0.4115
1383	1.49E-05	8.88E+11	-2.65E-03	-2.58E-03	-2.50E-03	-2.41E-03	0.3430	0.3354	0.3258	0.3140	0.4124	0.4121	0.4118	0.4113
1383	1.49E-05	9.25E+11	-2.62E-03	-2.55E-03	-2.47E-03	-2.38E-03	0.3382	0.3307	0.3213	0.3096	0.4122	0.4119	0.4116	0.4112
1383	1.49E-05	9.63E+11	-2.58E-03	-2.52E-03	-2.44E-03	-2.34E-03	0.3337	0.3263	0.3170	0.3055	0.4121	0.4118	0.4114	0.4110
1383	1.49E-05	1.00E+12	-2.55E-03	-2.48E-03	-2.41E-03	-2.31E-03	0.3295	0.3222	0.3129	0.3016	0.4119	0.4116	0.4113	0.4109

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1490	1.38E-05	1.00E+11	-5.67E-03	-5.52E-03	-5.35E-03	-5.15E-03	0.7437	0.7267	0.7054	0.6792	0.4276	0.4269	0.4261	0.4251
1490	1.38E-05	1.38E+11	-5.09E-03	-4.96E-03	-4.80E-03	-4.62E-03	0.6653	0.6503	0.6313	0.6080	0.4245	0.4240	0.4233	0.4224
1490	1.38E-05	1.75E+11	-4.69E-03	-4.57E-03	-4.43E-03	-4.26E-03	0.6120	0.5982	0.5808	0.5594	0.4225	0.4220	0.4213	0.4205
1490	1.38E-05	2.13E+11	-4.40E-03	-4.28E-03	-4.15E-03	-3.99E-03	0.5723	0.5594	0.5432	0.5233	0.4210	0.4205	0.4199	0.4192
1490	1.38E-05	2.50E+11	-4.16E-03	-4.05E-03	-3.93E-03	-3.78E-03	0.5412	0.5291	0.5138	0.4950	0.4198	0.4194	0.4188	0.4181
1490	1.38E-05	2.88E+11	-3.97E-03	-3.87E-03	-3.75E-03	-3.60E-03	0.5159	0.5044	0.4898	0.4719	0.4189	0.4184	0.4179	0.4172
1490	1.38E-05	3.25E+11	-3.81E-03	-3.71E-03	-3.59E-03	-3.46E-03	0.4947	0.4837	0.4697	0.4526	0.4181	0.4177	0.4171	0.4165
1490	1.38E-05	3.63E+11	-3.67E-03	-3.58E-03	-3.46E-03	-3.33E-03	0.4766	0.4660	0.4526	0.4361	0.4174	0.4170	0.4165	0.4159
1490	1.38E-05	4.00E+11	-3.55E-03	-3.46E-03	-3.35E-03	-3.22E-03	0.4609	0.4506	0.4376	0.4216	0.4168	0.4164	0.4159	0.4153
1490	1.38E-05	4.38E+11	-3.45E-03	-3.36E-03	-3.25E-03	-3.13E-03	0.4470	0.4370	0.4245	0.4091	0.4163	0.4159	0.4154	0.4149
1490	1.38E-05	4.75E+11	-3.36E-03	-3.27E-03	-3.16E-03	-3.04E-03	0.4348	0.4251	0.4128	0.3978	0.4158	0.4155	0.4150	0.4144
1490	1.38E-05	5.13E+11	-3.27E-03	-3.19E-03	-3.09E-03	-2.97E-03	0.4237	0.4142	0.4023	0.3877	0.4154	0.4150	0.4146	0.4141
1490	1.38E-05	5.50E+11	-3.19E-03	-3.11E-03	-3.01E-03	-2.90E-03	0.4137	0.4044	0.3928	0.3786	0.4150	0.4147	0.4143	0.4137
1490	1.38E-05	5.88E+11	-3.12E-03	-3.04E-03	-2.95E-03	-2.83E-03	0.4045	0.3955	0.3842	0.3701	0.4147	0.4144	0.4139	0.4134
1490	1.38E-05	6.25E+11	-3.06E-03	-2.98E-03	-2.89E-03	-2.78E-03	0.3960	0.3872	0.3761	0.3624	0.4144	0.4140	0.4136	0.4131
1490	1.38E-05	6.63E+11	-3.00E-03	-2.92E-03	-2.83E-03	-2.72E-03	0.3882	0.3796	0.3687	0.3554	0.4141	0.4138	0.4134	0.4129
1490	1.38E-05	7.00E+11	-2.95E-03	-2.87E-03	-2.78E-03	-2.67E-03	0.3811	0.3726	0.3619	0.3488	0.4138	0.4135	0.4131	0.4126
1490	1.38E-05	7.38E+11	-2.90E-03	-2.82E-03	-2.73E-03	-2.63E-03	0.3744	0.3661	0.3556	0.3427	0.4136	0.4133	0.4129	0.4124
1490	1.38E-05	7.75E+11	-2.85E-03	-2.77E-03	-2.69E-03	-2.58E-03	0.3682	0.3600	0.3497	0.3370	0.4133	0.4130	0.4127	0.4122
1490	1.38E-05	8.13E+11	-2.80E-03	-2.73E-03	-2.64E-03	-2.54E-03	0.3624	0.3543	0.3442	0.3317	0.4131	0.4128	0.4124	0.4120
1490	1.38E-05	8.50E+11	-2.76E-03	-2.69E-03	-2.60E-03	-2.51E-03	0.3569	0.3490	0.3390	0.3267	0.4129	0.4126	0.4123	0.4118
1490	1.38E-05	8.88E+11	-2.72E-03	-2.65E-03	-2.57E-03	-2.47E-03	0.3517	0.3439	0.3341	0.3220	0.4127	0.4124	0.4121	0.4116
1490	1.38E-05	9.25E+11	-2.68E-03	-2.61E-03	-2.53E-03	-2.44E-03	0.3468	0.3391	0.3294	0.3175	0.4125	0.4123	0.4119	0.4115
1490	1.38E-05	9.63E+11	-2.65E-03	-2.58E-03	-2.50E-03	-2.40E-03	0.3422	0.3346	0.3251	0.3133	0.4124	0.4121	0.4117	0.4113
1490	1.38E-05	1.00E+12	-2.62E-03	-2.55E-03	-2.47E-03	-2.37E-03	0.3378	0.3303	0.3209	0.3093	0.4122	0.4119	0.4116	0.4112
1598	1.29E-05	1.00E+11	-5.81E-03	-5.65E-03	-5.48E-03	-5.27E-03	0.7623	0.7448	0.7229	0.6960	0.4283	0.4276	0.4268	0.4257
1598	1.29E-05	1.38E+11	-5.21E-03	-5.08E-03	-4.92E-03	-4.73E-03	0.6818	0.6663	0.6468	0.6229	0.4252	0.4246	0.4238	0.4229
1598	1.29E-05	1.75E+11	-4.81E-03	-4.68E-03	-4.53E-03	-4.36E-03	0.6270	0.6128	0.5950	0.5731	0.4231	0.4225	0.4219	0.4210

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1598	1.29E-05	2.13E+11	-4.50E-03	-4.38E-03	-4.25E-03	-4.08E-03	0.5863	0.5731	0.5564	0.5360	0.4215	0.4210	0.4204	0.4196
1598	1.29E-05	2.50E+11	-4.26E-03	-4.15E-03	-4.02E-03	-3.87E-03	0.5544	0.5419	0.5262	0.5070	0.4203	0.4199	0.4193	0.4185
1598	1.29E-05	2.88E+11	-4.07E-03	-3.96E-03	-3.83E-03	-3.69E-03	0.5284	0.5166	0.5016	0.4833	0.4193	0.4189	0.4183	0.4176
1598	1.29E-05	3.25E+11	-3.90E-03	-3.80E-03	-3.68E-03	-3.54E-03	0.5067	0.4954	0.4811	0.4635	0.4185	0.4181	0.4176	0.4169
1598	1.29E-05	3.63E+11	-3.76E-03	-3.66E-03	-3.55E-03	-3.41E-03	0.4882	0.4771	0.4635	0.4466	0.4178	0.4174	0.4169	0.4163
1598	1.29E-05	4.00E+11	-3.64E-03	-3.54E-03	-3.43E-03	-3.30E-03	0.4720	0.4615	0.4482	0.4319	0.4172	0.4168	0.4163	0.4157
1598	1.29E-05	4.38E+11	-3.53E-03	-3.44E-03	-3.33E-03	-3.20E-03	0.4579	0.4476	0.4347	0.4189	0.4167	0.4163	0.4158	0.4152
1598	1.29E-05	4.75E+11	-3.44E-03	-3.35E-03	-3.24E-03	-3.12E-03	0.4452	0.4353	0.4228	0.4074	0.4162	0.4158	0.4154	0.4148
1598	1.29E-05	5.13E+11	-3.35E-03	-3.26E-03	-3.16E-03	-3.04E-03	0.4339	0.4242	0.4120	0.3970	0.4158	0.4154	0.4150	0.4144
1598	1.29E-05	5.50E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.97E-03	0.4236	0.4141	0.4023	0.3876	0.4154	0.4150	0.4146	0.4141
1598	1.29E-05	5.88E+11	-3.20E-03	-3.12E-03	-3.02E-03	-2.90E-03	0.4142	0.4050	0.3934	0.3791	0.4150	0.4147	0.4143	0.4137
1598	1.29E-05	6.25E+11	-3.13E-03	-3.05E-03	-2.96E-03	-2.84E-03	0.4056	0.3966	0.3852	0.3712	0.4147	0.4144	0.4140	0.4134
1598	1.29E-05	6.63E+11	-3.07E-03	-2.99E-03	-2.90E-03	-2.79E-03	0.3976	0.3887	0.3776	0.3639	0.4144	0.4141	0.4137	0.4132
1598	1.29E-05	7.00E+11	-3.02E-03	-2.94E-03	-2.84E-03	-2.74E-03	0.3902	0.3815	0.3706	0.3571	0.4142	0.4138	0.4134	0.4129
1598	1.29E-05	7.38E+11	-2.96E-03	-2.89E-03	-2.80E-03	-2.69E-03	0.3834	0.3749	0.3641	0.3509	0.4139	0.4136	0.4132	0.4127
1598	1.29E-05	7.75E+11	-2.92E-03	-2.84E-03	-2.75E-03	-2.64E-03	0.3770	0.3686	0.3581	0.3451	0.4137	0.4134	0.4130	0.4125
1598	1.29E-05	8.13E+11	-2.87E-03	-2.79E-03	-2.71E-03	-2.60E-03	0.3710	0.3628	0.3524	0.3396	0.4134	0.4131	0.4128	0.4123
1598	1.29E-05	8.50E+11	-2.83E-03	-2.75E-03	-2.67E-03	-2.56E-03	0.3654	0.3573	0.3471	0.3345	0.4132	0.4129	0.4126	0.4121
1598	1.29E-05	8.88E+11	-2.79E-03	-2.71E-03	-2.63E-03	-2.53E-03	0.3601	0.3521	0.3421	0.3297	0.4130	0.4127	0.4124	0.4119
1598	1.29E-05	9.25E+11	-2.75E-03	-2.68E-03	-2.59E-03	-2.49E-03	0.3551	0.3472	0.3373	0.3251	0.4129	0.4126	0.4122	0.4117
1598	1.29E-05	9.63E+11	-2.71E-03	-2.64E-03	-2.56E-03	-2.46E-03	0.3504	0.3426	0.3328	0.3208	0.4127	0.4124	0.4120	0.4116
1598	1.29E-05	1.00E+12	-2.68E-03	-2.61E-03	-2.52E-03	-2.43E-03	0.3459	0.3382	0.3286	0.3167	0.4125	0.4122	0.4119	0.4114
1706	1.21E-05	1.00E+11	-5.94E-03	-5.78E-03	-5.60E-03	-5.39E-03	0.7801	0.7622	0.7397	0.7122	0.4290	0.4283	0.4274	0.4264
1706	1.21E-05	1.38E+11	-5.33E-03	-5.19E-03	-5.03E-03	-4.84E-03	0.6975	0.6816	0.6617	0.6372	0.4258	0.4252	0.4244	0.4235
1706	1.21E-05	1.75E+11	-4.91E-03	-4.78E-03	-4.63E-03	-4.46E-03	0.6414	0.6268	0.6086	0.5862	0.4236	0.4231	0.4224	0.4215
1706	1.21E-05	2.13E+11	-4.60E-03	-4.48E-03	-4.34E-03	-4.18E-03	0.5997	0.5861	0.5691	0.5482	0.4220	0.4215	0.4209	0.4201
1706	1.21E-05	2.50E+11	-4.36E-03	-4.24E-03	-4.11E-03	-3.95E-03	0.5670	0.5542	0.5382	0.5185	0.4208	0.4203	0.4197	0.4190
1706	1.21E-05	2.88E+11	-4.16E-03	-4.05E-03	-3.92E-03	-3.77E-03	0.5404	0.5283	0.5130	0.4942	0.4198	0.4193	0.4188	0.4181

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1706	1.21E-05	3.25E+11	-3.99E-03	-3.88E-03	-3.76E-03	-3.62E-03	0.5182	0.5066	0.4919	0.4740	0.4190	0.4185	0.4180	0.4173
1706	1.21E-05	3.63E+11	-3.84E-03	-3.74E-03	-3.63E-03	-3.49E-03	0.4992	0.4880	0.4739	0.4566	0.4182	0.4178	0.4173	0.4166
1706	1.21E-05	4.00E+11	-3.72E-03	-3.62E-03	-3.51E-03	-3.37E-03	0.4827	0.4718	0.4583	0.4416	0.4176	0.4172	0.4167	0.4161
1706	1.21E-05	4.38E+11	-3.61E-03	-3.51E-03	-3.40E-03	-3.27E-03	0.4682	0.4577	0.4445	0.4283	0.4171	0.4167	0.4162	0.4156
1706	1.21E-05	4.75E+11	-3.51E-03	-3.42E-03	-3.31E-03	-3.19E-03	0.4552	0.4451	0.4323	0.4165	0.4166	0.4162	0.4157	0.4151
1706	1.21E-05	5.13E+11	-3.42E-03	-3.33E-03	-3.23E-03	-3.11E-03	0.4436	0.4337	0.4212	0.4059	0.4161	0.4158	0.4153	0.4147
1706	1.21E-05	5.50E+11	-3.34E-03	-3.26E-03	-3.15E-03	-3.03E-03	0.4331	0.4234	0.4113	0.3963	0.4158	0.4154	0.4149	0.4144
1706	1.21E-05	5.88E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.97E-03	0.4235	0.4141	0.4022	0.3876	0.4154	0.4150	0.4146	0.4141
1706	1.21E-05	6.25E+11	-3.20E-03	-3.12E-03	-3.02E-03	-2.91E-03	0.4147	0.4055	0.3938	0.3795	0.4151	0.4147	0.4143	0.4138
1706	1.21E-05	6.63E+11	-3.14E-03	-3.06E-03	-2.96E-03	-2.85E-03	0.4066	0.3975	0.3861	0.3721	0.4148	0.4144	0.4140	0.4135
1706	1.21E-05	7.00E+11	-3.08E-03	-3.00E-03	-2.91E-03	-2.80E-03	0.3989	0.3901	0.3789	0.3651	0.4145	0.4142	0.4137	0.4132
1706	1.21E-05	7.38E+11	-3.03E-03	-2.95E-03	-2.86E-03	-2.75E-03	0.3920	0.3832	0.3723	0.3587	0.4142	0.4139	0.4135	0.4130
1706	1.21E-05	7.75E+11	-2.98E-03	-2.90E-03	-2.81E-03	-2.70E-03	0.3854	0.3769	0.3661	0.3528	0.4140	0.4137	0.4133	0.4128
1706	1.21E-05	8.13E+11	-2.93E-03	-2.86E-03	-2.77E-03	-2.66E-03	0.3793	0.3709	0.3603	0.3472	0.4138	0.4134	0.4130	0.4126
1706	1.21E-05	8.50E+11	-2.89E-03	-2.81E-03	-2.72E-03	-2.62E-03	0.3736	0.3653	0.3548	0.3420	0.4135	0.4132	0.4128	0.4124
1706	1.21E-05	8.88E+11	-2.85E-03	-2.77E-03	-2.69E-03	-2.58E-03	0.3682	0.3600	0.3497	0.3370	0.4133	0.4130	0.4127	0.4122
1706	1.21E-05	9.25E+11	-2.81E-03	-2.73E-03	-2.65E-03	-2.55E-03	0.3631	0.3550	0.3448	0.3323	0.4131	0.4128	0.4125	0.4120
1706	1.21E-05	9.63E+11	-2.77E-03	-2.70E-03	-2.61E-03	-2.51E-03	0.3582	0.3503	0.3402	0.3279	0.4130	0.4127	0.4123	0.4118
1706	1.21E-05	1.00E+12	-2.74E-03	-2.66E-03	-2.58E-03	-2.48E-03	0.3536	0.3458	0.3359	0.3237	0.4128	0.4125	0.4121	0.4117
1814	1.14E-05	1.00E+11	-6.06E-03	-5.90E-03	-5.72E-03	-5.50E-03	0.7972	0.7789	0.7559	0.7277	0.4296	0.4289	0.4280	0.4270
1814	1.14E-05	1.38E+11	-5.44E-03	-5.30E-03	-5.13E-03	-4.94E-03	0.7126	0.6964	0.6760	0.6510	0.4264	0.4257	0.4250	0.4240
1814	1.14E-05	1.75E+11	-5.02E-03	-4.88E-03	-4.73E-03	-4.55E-03	0.6552	0.6403	0.6217	0.5987	0.4242	0.4236	0.4229	0.4220
1814	1.14E-05	2.13E+11	-4.70E-03	-4.57E-03	-4.43E-03	-4.26E-03	0.6125	0.5987	0.5813	0.5599	0.4225	0.4220	0.4213	0.4205
1814	1.14E-05	2.50E+11	-4.45E-03	-4.33E-03	-4.19E-03	-4.04E-03	0.5791	0.5661	0.5497	0.5295	0.4213	0.4208	0.4201	0.4194
1814	1.14E-05	2.88E+11	-4.24E-03	-4.13E-03	-4.00E-03	-3.85E-03	0.5519	0.5395	0.5239	0.5047	0.4202	0.4198	0.4192	0.4184
1814	1.14E-05	3.25E+11	-4.07E-03	-3.96E-03	-3.84E-03	-3.69E-03	0.5292	0.5173	0.5024	0.4840	0.4194	0.4189	0.4184	0.4177
1814	1.14E-05	3.63E+11	-3.92E-03	-3.82E-03	-3.70E-03	-3.56E-03	0.5098	0.4983	0.4840	0.4663	0.4186	0.4182	0.4177	0.4170
1814	1.14E-05	4.00E+11	-3.80E-03	-3.70E-03	-3.58E-03	-3.45E-03	0.4929	0.4817	0.4680	0.4509	0.4180	0.4176	0.4171	0.4164

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1814	1.14E-05	4.38E+11	-3.68E-03	-3.59E-03	-3.48E-03	-3.34E-03	0.4781	0.4674	0.4539	0.4374	0.4174	0.4170	0.4165	0.4159
1814	1.14E-05	4.75E+11	-3.58E-03	-3.49E-03	-3.38E-03	-3.25E-03	0.4648	0.4545	0.4414	0.4253	0.4169	0.4166	0.4161	0.4155
1814	1.14E-05	5.13E+11	-3.49E-03	-3.40E-03	-3.30E-03	-3.17E-03	0.4530	0.4429	0.4301	0.4145	0.4165	0.4161	0.4156	0.4151
1814	1.14E-05	5.50E+11	-3.41E-03	-3.32E-03	-3.22E-03	-3.10E-03	0.4422	0.4324	0.4199	0.4046	0.4161	0.4157	0.4153	0.4147
1814	1.14E-05	5.88E+11	-3.34E-03	-3.25E-03	-3.15E-03	-3.03E-03	0.4324	0.4228	0.4106	0.3957	0.4157	0.4154	0.4149	0.4144
1814	1.14E-05	6.25E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.97E-03	0.4234	0.4140	0.4021	0.3875	0.4154	0.4150	0.4146	0.4141
1814	1.14E-05	6.63E+11	-3.21E-03	-3.12E-03	-3.02E-03	-2.91E-03	0.4152	0.4059	0.3942	0.3799	0.4151	0.4147	0.4143	0.4138
1814	1.14E-05	7.00E+11	-3.15E-03	-3.06E-03	-2.97E-03	-2.86E-03	0.4075	0.3984	0.3870	0.3729	0.4148	0.4145	0.4140	0.4135
1814	1.14E-05	7.38E+11	-3.09E-03	-3.01E-03	-2.92E-03	-2.81E-03	0.4002	0.3914	0.3801	0.3663	0.4145	0.4142	0.4138	0.4133
1814	1.14E-05	7.75E+11	-3.04E-03	-2.96E-03	-2.87E-03	-2.76E-03	0.3935	0.3848	0.3737	0.3602	0.4143	0.4140	0.4135	0.4130
1814	1.14E-05	8.13E+11	-2.99E-03	-2.92E-03	-2.82E-03	-2.72E-03	0.3873	0.3787	0.3678	0.3545	0.4140	0.4137	0.4133	0.4128
1814	1.14E-05	8.50E+11	-2.95E-03	-2.87E-03	-2.78E-03	-2.68E-03	0.3814	0.3729	0.3623	0.3491	0.4138	0.4135	0.4131	0.4126
1814	1.14E-05	8.88E+11	-2.91E-03	-2.83E-03	-2.74E-03	-2.64E-03	0.3759	0.3675	0.3570	0.3441	0.4136	0.4133	0.4129	0.4124
1814	1.14E-05	9.25E+11	-2.87E-03	-2.79E-03	-2.70E-03	-2.60E-03	0.3707	0.3624	0.3521	0.3393	0.4134	0.4131	0.4127	0.4123
1814	1.14E-05	9.63E+11	-2.83E-03	-2.75E-03	-2.67E-03	-2.57E-03	0.3657	0.3576	0.3474	0.3348	0.4132	0.4129	0.4126	0.4121
1814	1.14E-05	1.00E+12	-2.79E-03	-2.72E-03	-2.63E-03	-2.53E-03	0.3610	0.3530	0.3429	0.3305	0.4131	0.4128	0.4124	0.4119
1922	1.07E-05	1.00E+11	-6.18E-03	-6.02E-03	-5.83E-03	-5.61E-03	0.8137	0.7950	0.7715	0.7427	0.4303	0.4295	0.4286	0.4275
1922	1.07E-05	1.38E+11	-5.55E-03	-5.40E-03	-5.23E-03	-5.04E-03	0.7272	0.7106	0.6898	0.6642	0.4269	0.4263	0.4255	0.4245
1922	1.07E-05	1.75E+11	-5.11E-03	-4.98E-03	-4.82E-03	-4.64E-03	0.6685	0.6533	0.6342	0.6108	0.4247	0.4241	0.4234	0.4225
1922	1.07E-05	2.13E+11	-4.79E-03	-4.66E-03	-4.52E-03	-4.35E-03	0.6249	0.6107	0.5930	0.5712	0.4230	0.4225	0.4218	0.4210
1922	1.07E-05	2.50E+11	-4.53E-03	-4.42E-03	-4.28E-03	-4.11E-03	0.5908	0.5775	0.5607	0.5401	0.4217	0.4212	0.4206	0.4198
1922	1.07E-05	2.88E+11	-4.33E-03	-4.21E-03	-4.08E-03	-3.93E-03	0.5630	0.5503	0.5344	0.5148	0.4207	0.4202	0.4196	0.4188
1922	1.07E-05	3.25E+11	-4.15E-03	-4.04E-03	-3.92E-03	-3.77E-03	0.5398	0.5277	0.5124	0.4937	0.4198	0.4193	0.4187	0.4180
1922	1.07E-05	3.63E+11	-4.00E-03	-3.90E-03	-3.77E-03	-3.63E-03	0.5200	0.5083	0.4936	0.4756	0.4190	0.4186	0.4180	0.4174
1922	1.07E-05	4.00E+11	-3.87E-03	-3.77E-03	-3.65E-03	-3.51E-03	0.5027	0.4915	0.4773	0.4599	0.4184	0.4179	0.4174	0.4168
1922	1.07E-05	4.38E+11	-3.76E-03	-3.66E-03	-3.54E-03	-3.41E-03	0.4876	0.4767	0.4629	0.4460	0.4178	0.4174	0.4169	0.4162
1922	1.07E-05	4.75E+11	-3.65E-03	-3.56E-03	-3.45E-03	-3.32E-03	0.4741	0.4635	0.4501	0.4337	0.4173	0.4169	0.4164	0.4158
1922	1.07E-05	5.13E+11	-3.56E-03	-3.47E-03	-3.36E-03	-3.23E-03	0.4620	0.4517	0.4387	0.4227	0.4168	0.4165	0.4160	0.4154

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
1922	1.07E-05	5.50E+11	-3.48E-03	-3.39E-03	-3.28E-03	-3.16E-03	0.4510	0.4409	0.4283	0.4127	0.4164	0.4160	0.4156	0.4150
1922	1.07E-05	5.88E+11	-3.40E-03	-3.31E-03	-3.21E-03	-3.09E-03	0.4410	0.4312	0.4188	0.4035	0.4161	0.4157	0.4152	0.4147
1922	1.07E-05	6.25E+11	-3.33E-03	-3.25E-03	-3.14E-03	-3.02E-03	0.4318	0.4222	0.4101	0.3952	0.4157	0.4153	0.4149	0.4143
1922	1.07E-05	6.63E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.97E-03	0.4234	0.4139	0.4021	0.3874	0.4154	0.4150	0.4146	0.4141
1922	1.07E-05	7.00E+11	-3.21E-03	-3.12E-03	-3.03E-03	-2.91E-03	0.4155	0.4063	0.3946	0.3803	0.4151	0.4148	0.4143	0.4138
1922	1.07E-05	7.38E+11	-3.15E-03	-3.07E-03	-2.97E-03	-2.86E-03	0.4083	0.3992	0.3877	0.3736	0.4148	0.4145	0.4141	0.4135
1922	1.07E-05	7.75E+11	-3.10E-03	-3.02E-03	-2.92E-03	-2.81E-03	0.4014	0.3925	0.3813	0.3673	0.4146	0.4142	0.4138	0.4133
1922	1.07E-05	8.13E+11	-3.05E-03	-2.97E-03	-2.88E-03	-2.77E-03	0.3949	0.3862	0.3751	0.3615	0.4143	0.4140	0.4136	0.4131
1922	1.07E-05	8.50E+11	-3.01E-03	-2.93E-03	-2.84E-03	-2.73E-03	0.3890	0.3803	0.3694	0.3560	0.4141	0.4138	0.4134	0.4129
1922	1.07E-05	8.88E+11	-2.96E-03	-2.89E-03	-2.79E-03	-2.69E-03	0.3833	0.3748	0.3641	0.3508	0.4139	0.4136	0.4132	0.4127
1922	1.07E-05	9.25E+11	-2.92E-03	-2.85E-03	-2.76E-03	-2.65E-03	0.3780	0.3696	0.3590	0.3460	0.4137	0.4134	0.4130	0.4125
1922	1.07E-05	9.63E+11	-2.88E-03	-2.81E-03	-2.72E-03	-2.62E-03	0.3729	0.3647	0.3542	0.3414	0.4135	0.4132	0.4128	0.4123
1922	1.07E-05	1.00E+12	-2.85E-03	-2.77E-03	-2.69E-03	-2.58E-03	0.3682	0.3600	0.3497	0.3370	0.4133	0.4130	0.4126	0.4122
2030	1.01E-05	1.00E+11	-6.30E-03	-6.13E-03	-5.94E-03	-5.72E-03	0.8296	0.8105	0.7866	0.7572	0.4309	0.4301	0.4292	0.4281
2030	1.01E-05	1.38E+11	-5.65E-03	-5.50E-03	-5.33E-03	-5.13E-03	0.7413	0.7244	0.7031	0.6770	0.4275	0.4268	0.4260	0.4250
2030	1.01E-05	1.75E+11	-5.21E-03	-5.07E-03	-4.91E-03	-4.73E-03	0.6813	0.6658	0.6464	0.6225	0.4252	0.4246	0.4238	0.4229
2030	1.01E-05	2.13E+11	-4.88E-03	-4.75E-03	-4.60E-03	-4.43E-03	0.6368	0.6224	0.6043	0.5821	0.4235	0.4229	0.4222	0.4214
2030	1.01E-05	2.50E+11	-4.62E-03	-4.50E-03	-4.36E-03	-4.19E-03	0.6020	0.5884	0.5713	0.5504	0.4221	0.4216	0.4210	0.4202
2030	1.01E-05	2.88E+11	-4.41E-03	-4.29E-03	-4.16E-03	-4.00E-03	0.5737	0.5608	0.5445	0.5246	0.4211	0.4206	0.4200	0.4192
2030	1.01E-05	3.25E+11	-4.23E-03	-4.12E-03	-3.99E-03	-3.84E-03	0.5500	0.5377	0.5221	0.5030	0.4202	0.4197	0.4191	0.4184
2030	1.01E-05	3.63E+11	-4.08E-03	-3.97E-03	-3.84E-03	-3.70E-03	0.5298	0.5179	0.5029	0.4845	0.4194	0.4189	0.4184	0.4177
2030	1.01E-05	4.00E+11	-3.94E-03	-3.84E-03	-3.72E-03	-3.58E-03	0.5122	0.5007	0.4863	0.4685	0.4187	0.4183	0.4178	0.4171
2030	1.01E-05	4.38E+11	-3.83E-03	-3.73E-03	-3.61E-03	-3.47E-03	0.4968	0.4856	0.4716	0.4544	0.4181	0.4177	0.4172	0.4166
2030	1.01E-05	4.75E+11	-3.72E-03	-3.62E-03	-3.51E-03	-3.38E-03	0.4830	0.4722	0.4586	0.4419	0.4176	0.4172	0.4167	0.4161
2030	1.01E-05	5.13E+11	-3.63E-03	-3.53E-03	-3.42E-03	-3.29E-03	0.4707	0.4601	0.4469	0.4306	0.4172	0.4168	0.4163	0.4157
2030	1.01E-05	5.50E+11	-3.54E-03	-3.45E-03	-3.34E-03	-3.22E-03	0.4595	0.4492	0.4363	0.4204	0.4167	0.4164	0.4159	0.4153
2030	1.01E-05	5.88E+11	-3.47E-03	-3.38E-03	-3.27E-03	-3.14E-03	0.4493	0.4393	0.4266	0.4111	0.4164	0.4160	0.4155	0.4149
2030	1.01E-05	6.25E+11	-3.39E-03	-3.31E-03	-3.20E-03	-3.08E-03	0.4399	0.4301	0.4178	0.4025	0.4160	0.4156	0.4152	0.4146

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2030	1.01E-05	6.63E+11	-3.33E-03	-3.24E-03	-3.14E-03	-3.02E-03	0.4313	0.4217	0.4096	0.3947	0.4157	0.4153	0.4149	0.4143
2030	1.01E-05	7.00E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.97E-03	0.4233	0.4139	0.4020	0.3874	0.4154	0.4150	0.4146	0.4140
2030	1.01E-05	7.38E+11	-3.21E-03	-3.13E-03	-3.03E-03	-2.91E-03	0.4159	0.4066	0.3950	0.3806	0.4151	0.4148	0.4143	0.4138
2030	1.01E-05	7.75E+11	-3.16E-03	-3.08E-03	-2.98E-03	-2.87E-03	0.4090	0.3998	0.3884	0.3743	0.4149	0.4145	0.4141	0.4136
2030	1.01E-05	8.13E+11	-3.11E-03	-3.03E-03	-2.93E-03	-2.82E-03	0.4025	0.3935	0.3822	0.3682	0.4146	0.4143	0.4139	0.4133
2030	1.01E-05	8.50E+11	-3.06E-03	-2.98E-03	-2.89E-03	-2.78E-03	0.3962	0.3874	0.3763	0.3626	0.4144	0.4141	0.4136	0.4131
2030	1.01E-05	8.88E+11	-3.02E-03	-2.94E-03	-2.85E-03	-2.74E-03	0.3905	0.3818	0.3708	0.3574	0.4142	0.4138	0.4134	0.4129
2030	1.01E-05	9.25E+11	-2.98E-03	-2.90E-03	-2.81E-03	-2.70E-03	0.3850	0.3765	0.3657	0.3524	0.4140	0.4136	0.4132	0.4128
2030	1.01E-05	9.63E+11	-2.94E-03	-2.86E-03	-2.77E-03	-2.67E-03	0.3799	0.3715	0.3608	0.3477	0.4138	0.4135	0.4131	0.4126
2030	1.01E-05	1.00E+12	-2.90E-03	-2.82E-03	-2.74E-03	-2.63E-03	0.3750	0.3667	0.3562	0.3433	0.4136	0.4133	0.4129	0.4124
2137	9.64E-06	1.00E+11	-6.41E-03	-6.24E-03	-6.04E-03	-5.82E-03	0.8450	0.8255	0.8010	0.7711	0.4315	0.4307	0.4298	0.4286
2137	9.64E-06	1.38E+11	-5.75E-03	-5.60E-03	-5.42E-03	-5.22E-03	0.7549	0.7376	0.7159	0.6893	0.4280	0.4273	0.4265	0.4255
2137	9.64E-06	1.75E+11	-5.30E-03	-5.16E-03	-5.00E-03	-4.81E-03	0.6936	0.6778	0.6580	0.6337	0.4256	0.4250	0.4243	0.4233
2137	9.64E-06	2.13E+11	-4.96E-03	-4.83E-03	-4.68E-03	-4.50E-03	0.6483	0.6336	0.6151	0.5925	0.4239	0.4233	0.4226	0.4218
2137	9.64E-06	2.50E+11	-4.70E-03	-4.58E-03	-4.43E-03	-4.26E-03	0.6128	0.5990	0.5816	0.5602	0.4225	0.4220	0.4214	0.4205
2137	9.64E-06	2.88E+11	-4.48E-03	-4.37E-03	-4.23E-03	-4.07E-03	0.5839	0.5708	0.5542	0.5339	0.4214	0.4209	0.4203	0.4196
2137	9.64E-06	3.25E+11	-4.30E-03	-4.19E-03	-4.06E-03	-3.90E-03	0.5598	0.5472	0.5314	0.5119	0.4205	0.4201	0.4195	0.4187
2137	9.64E-06	3.63E+11	-4.15E-03	-4.04E-03	-3.91E-03	-3.76E-03	0.5392	0.5271	0.5118	0.4931	0.4197	0.4193	0.4187	0.4180
2137	9.64E-06	4.00E+11	-4.01E-03	-3.91E-03	-3.78E-03	-3.64E-03	0.5213	0.5096	0.4949	0.4768	0.4191	0.4186	0.4181	0.4174
2137	9.64E-06	4.38E+11	-3.89E-03	-3.79E-03	-3.67E-03	-3.53E-03	0.5056	0.4942	0.4800	0.4625	0.4185	0.4181	0.4175	0.4169
2137	9.64E-06	4.75E+11	-3.79E-03	-3.69E-03	-3.57E-03	-3.44E-03	0.4916	0.4806	0.4667	0.4497	0.4180	0.4175	0.4170	0.4164
2137	9.64E-06	5.13E+11	-3.69E-03	-3.59E-03	-3.48E-03	-3.35E-03	0.4790	0.4683	0.4548	0.4382	0.4175	0.4171	0.4166	0.4159
2137	9.64E-06	5.50E+11	-3.61E-03	-3.51E-03	-3.40E-03	-3.27E-03	0.4676	0.4571	0.4440	0.4278	0.4170	0.4167	0.4162	0.4156
2137	9.64E-06	5.88E+11	-3.53E-03	-3.43E-03	-3.33E-03	-3.20E-03	0.4572	0.4470	0.4341	0.4183	0.4167	0.4163	0.4158	0.4152
2137	9.64E-06	6.25E+11	-3.45E-03	-3.36E-03	-3.26E-03	-3.13E-03	0.4477	0.4377	0.4251	0.4096	0.4163	0.4159	0.4155	0.4149
2137	9.64E-06	6.63E+11	-3.39E-03	-3.30E-03	-3.19E-03	-3.07E-03	0.4389	0.4291	0.4168	0.4016	0.4160	0.4156	0.4151	0.4146
2137	9.64E-06	7.00E+11	-3.33E-03	-3.24E-03	-3.14E-03	-3.02E-03	0.4308	0.4212	0.4091	0.3942	0.4157	0.4153	0.4149	0.4143
2137	9.64E-06	7.38E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4232	0.4138	0.4019	0.3873	0.4154	0.4150	0.4146	0.4140

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2137	9.64E-06	7.75E+11	-3.21E-03	-3.13E-03	-3.03E-03	-2.92E-03	0.4161	0.4069	0.3952	0.3808	0.4151	0.4148	0.4143	0.4138
2137	9.64E-06	8.13E+11	-3.16E-03	-3.08E-03	-2.98E-03	-2.87E-03	0.4095	0.4004	0.3889	0.3748	0.4149	0.4145	0.4141	0.4136
2137	9.64E-06	8.50E+11	-3.12E-03	-3.03E-03	-2.94E-03	-2.83E-03	0.4033	0.3943	0.3830	0.3690	0.4146	0.4143	0.4139	0.4134
2137	9.64E-06	8.88E+11	-3.07E-03	-2.99E-03	-2.90E-03	-2.79E-03	0.3973	0.3885	0.3773	0.3636	0.4144	0.4141	0.4137	0.4132
2137	9.64E-06	9.25E+11	-3.03E-03	-2.95E-03	-2.86E-03	-2.75E-03	0.3918	0.3831	0.3721	0.3586	0.4142	0.4139	0.4135	0.4130
2137	9.64E-06	9.63E+11	-2.99E-03	-2.91E-03	-2.82E-03	-2.71E-03	0.3866	0.3780	0.3671	0.3538	0.4140	0.4137	0.4133	0.4128
2137	9.64E-06	1.00E+12	-2.95E-03	-2.87E-03	-2.78E-03	-2.68E-03	0.3816	0.3731	0.3624	0.3493	0.4138	0.4135	0.4131	0.4126
2245	9.17E-06	1.00E+11	-6.52E-03	-6.35E-03	-6.15E-03	-5.91E-03	0.8600	0.8401	0.8152	0.7847	0.4320	0.4313	0.4303	0.4291
2245	9.17E-06	1.38E+11	-5.85E-03	-5.70E-03	-5.52E-03	-5.31E-03	0.7681	0.7505	0.7284	0.7013	0.4285	0.4278	0.4270	0.4259
2245	9.17E-06	1.75E+11	-5.39E-03	-5.25E-03	-5.08E-03	-4.89E-03	0.7057	0.6896	0.6694	0.6447	0.4261	0.4255	0.4247	0.4238
2245	9.17E-06	2.13E+11	-5.05E-03	-4.92E-03	-4.76E-03	-4.58E-03	0.6595	0.6445	0.6257	0.6027	0.4243	0.4238	0.4230	0.4222
2245	9.17E-06	2.50E+11	-4.78E-03	-4.65E-03	-4.51E-03	-4.34E-03	0.6234	0.6093	0.5915	0.5698	0.4229	0.4224	0.4217	0.4209
2245	9.17E-06	2.88E+11	-4.56E-03	-4.44E-03	-4.30E-03	-4.14E-03	0.5940	0.5806	0.5637	0.5430	0.4218	0.4213	0.4207	0.4199
2245	9.17E-06	3.25E+11	-4.37E-03	-4.26E-03	-4.13E-03	-3.97E-03	0.5694	0.5566	0.5404	0.5206	0.4209	0.4204	0.4198	0.4191
2245	9.17E-06	3.63E+11	-4.22E-03	-4.11E-03	-3.98E-03	-3.83E-03	0.5484	0.5361	0.5206	0.5015	0.4201	0.4196	0.4190	0.4183
2245	9.17E-06	4.00E+11	-4.08E-03	-3.97E-03	-3.85E-03	-3.70E-03	0.5302	0.5183	0.5033	0.4849	0.4194	0.4190	0.4184	0.4177
2245	9.17E-06	4.38E+11	-3.96E-03	-3.85E-03	-3.73E-03	-3.59E-03	0.5142	0.5026	0.4881	0.4703	0.4188	0.4184	0.4178	0.4172
2245	9.17E-06	4.75E+11	-3.85E-03	-3.75E-03	-3.63E-03	-3.49E-03	0.4999	0.4887	0.4746	0.4573	0.4183	0.4178	0.4173	0.4167
2245	9.17E-06	5.13E+11	-3.75E-03	-3.65E-03	-3.54E-03	-3.41E-03	0.4871	0.4762	0.4625	0.4456	0.4178	0.4174	0.4169	0.4162
2245	9.17E-06	5.50E+11	-3.67E-03	-3.57E-03	-3.46E-03	-3.33E-03	0.4755	0.4649	0.4515	0.4350	0.4173	0.4169	0.4164	0.4158
2245	9.17E-06	5.88E+11	-3.59E-03	-3.49E-03	-3.38E-03	-3.25E-03	0.4649	0.4546	0.4415	0.4254	0.4169	0.4166	0.4161	0.4155
2245	9.17E-06	6.25E+11	-3.51E-03	-3.42E-03	-3.31E-03	-3.19E-03	0.4553	0.4451	0.4323	0.4165	0.4166	0.4162	0.4157	0.4151
2245	9.17E-06	6.63E+11	-3.44E-03	-3.35E-03	-3.25E-03	-3.12E-03	0.4463	0.4364	0.4238	0.4084	0.4162	0.4159	0.4154	0.4148
2245	9.17E-06	7.00E+11	-3.38E-03	-3.29E-03	-3.19E-03	-3.07E-03	0.4380	0.4283	0.4160	0.4008	0.4159	0.4156	0.4151	0.4145
2245	9.17E-06	7.38E+11	-3.32E-03	-3.23E-03	-3.13E-03	-3.01E-03	0.4303	0.4207	0.4087	0.3938	0.4157	0.4153	0.4148	0.4143
2245	9.17E-06	7.75E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4232	0.4137	0.4018	0.3872	0.4154	0.4150	0.4146	0.4140
2245	9.17E-06	8.13E+11	-3.22E-03	-3.13E-03	-3.03E-03	-2.92E-03	0.4164	0.4072	0.3955	0.3811	0.4151	0.4148	0.4143	0.4138
2245	9.17E-06	8.50E+11	-3.17E-03	-3.08E-03	-2.99E-03	-2.87E-03	0.4101	0.4010	0.3895	0.3753	0.4149	0.4146	0.4141	0.4136

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2245	9.17E-06	8.88E+11	-3.12E-03	-3.04E-03	-2.94E-03	-2.83E-03	0.4042	0.3952	0.3838	0.3698	0.4147	0.4143	0.4139	0.4134
2245	9.17E-06	9.25E+11	-3.08E-03	-3.00E-03	-2.90E-03	-2.79E-03	0.3984	0.3895	0.3784	0.3646	0.4145	0.4141	0.4137	0.4132
2245	9.17E-06	9.63E+11	-3.04E-03	-2.96E-03	-2.87E-03	-2.76E-03	0.3931	0.3843	0.3733	0.3598	0.4143	0.4139	0.4135	0.4130
2245	9.17E-06	1.00E+12	-3.00E-03	-2.92E-03	-2.83E-03	-2.72E-03	0.3880	0.3794	0.3685	0.3551	0.4141	0.4138	0.4133	0.4129
2353	8.75E-06	1.00E+11	-6.62E-03	-6.45E-03	-6.25E-03	-6.01E-03	0.8742	0.8540	0.8287	0.7979	0.4326	0.4318	0.4309	0.4297
2353	8.75E-06	1.38E+11	-5.94E-03	-5.79E-03	-5.61E-03	-5.39E-03	0.7809	0.7630	0.7405	0.7130	0.4290	0.4283	0.4274	0.4264
2353	8.75E-06	1.75E+11	-5.48E-03	-5.33E-03	-5.17E-03	-4.97E-03	0.7174	0.7010	0.6805	0.6553	0.4265	0.4259	0.4251	0.4242
2353	8.75E-06	2.13E+11	-5.13E-03	-4.99E-03	-4.84E-03	-4.65E-03	0.6704	0.6552	0.6360	0.6126	0.4247	0.4242	0.4234	0.4225
2353	8.75E-06	2.50E+11	-4.85E-03	-4.73E-03	-4.58E-03	-4.41E-03	0.6336	0.6193	0.6012	0.5791	0.4233	0.4228	0.4221	0.4213
2353	8.75E-06	2.88E+11	-4.63E-03	-4.51E-03	-4.37E-03	-4.20E-03	0.6037	0.5900	0.5729	0.5519	0.4222	0.4217	0.4210	0.4202
2353	8.75E-06	3.25E+11	-4.44E-03	-4.33E-03	-4.19E-03	-4.03E-03	0.5787	0.5656	0.5492	0.5291	0.4212	0.4208	0.4201	0.4194
2353	8.75E-06	3.63E+11	-4.28E-03	-4.17E-03	-4.04E-03	-3.89E-03	0.5573	0.5448	0.5290	0.5096	0.4204	0.4200	0.4194	0.4186
2353	8.75E-06	4.00E+11	-4.14E-03	-4.04E-03	-3.91E-03	-3.76E-03	0.5388	0.5267	0.5115	0.4928	0.4197	0.4193	0.4187	0.4180
2353	8.75E-06	4.38E+11	-4.02E-03	-3.92E-03	-3.79E-03	-3.65E-03	0.5225	0.5108	0.4960	0.4779	0.4191	0.4187	0.4181	0.4174
2353	8.75E-06	4.75E+11	-3.91E-03	-3.81E-03	-3.69E-03	-3.55E-03	0.5080	0.4966	0.4823	0.4647	0.4186	0.4181	0.4176	0.4169
2353	8.75E-06	5.13E+11	-3.81E-03	-3.71E-03	-3.60E-03	-3.46E-03	0.4950	0.4839	0.4699	0.4528	0.4181	0.4177	0.4171	0.4165
2353	8.75E-06	5.50E+11	-3.72E-03	-3.63E-03	-3.51E-03	-3.38E-03	0.4832	0.4724	0.4588	0.4420	0.4176	0.4172	0.4167	0.4161
2353	8.75E-06	5.88E+11	-3.64E-03	-3.55E-03	-3.43E-03	-3.30E-03	0.4724	0.4619	0.4486	0.4322	0.4172	0.4168	0.4163	0.4157
2353	8.75E-06	6.25E+11	-3.57E-03	-3.47E-03	-3.36E-03	-3.24E-03	0.4626	0.4523	0.4392	0.4232	0.4169	0.4165	0.4160	0.4154
2353	8.75E-06	6.63E+11	-3.50E-03	-3.41E-03	-3.30E-03	-3.17E-03	0.4535	0.4434	0.4306	0.4149	0.4165	0.4161	0.4157	0.4151
2353	8.75E-06	7.00E+11	-3.43E-03	-3.34E-03	-3.24E-03	-3.12E-03	0.4451	0.4352	0.4226	0.4073	0.4162	0.4158	0.4154	0.4148
2353	8.75E-06	7.38E+11	-3.37E-03	-3.29E-03	-3.18E-03	-3.06E-03	0.4373	0.4275	0.4152	0.4001	0.4159	0.4155	0.4151	0.4145
2353	8.75E-06	7.75E+11	-3.32E-03	-3.23E-03	-3.13E-03	-3.01E-03	0.4300	0.4204	0.4083	0.3934	0.4156	0.4153	0.4148	0.4143
2353	8.75E-06	8.13E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4231	0.4137	0.4018	0.3872	0.4154	0.4150	0.4146	0.4140
2353	8.75E-06	8.50E+11	-3.22E-03	-3.13E-03	-3.03E-03	-2.92E-03	0.4167	0.4074	0.3957	0.3813	0.4151	0.4148	0.4144	0.4138
2353	8.75E-06	8.88E+11	-3.17E-03	-3.09E-03	-2.99E-03	-2.88E-03	0.4106	0.4015	0.3900	0.3758	0.4149	0.4146	0.4141	0.4136
2353	8.75E-06	9.25E+11	-3.13E-03	-3.05E-03	-2.95E-03	-2.84E-03	0.4049	0.3959	0.3845	0.3705	0.4147	0.4144	0.4139	0.4134
2353	8.75E-06	9.63E+11	-3.09E-03	-3.01E-03	-2.91E-03	-2.80E-03	0.3994	0.3905	0.3793	0.3655	0.4145	0.4142	0.4138	0.4132

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2353	8.75E-06	1.00E+12	-3.05E-03	-2.97E-03	-2.87E-03	-2.76E-03	0.3942	0.3855	0.3744	0.3608	0.4143	0.4140	0.4136	0.4131
2461	8.37E-06	1.00E+11	-6.72E-03	-6.55E-03	-6.34E-03	-6.10E-03	0.8887	0.8682	0.8421	0.8107	0.4332	0.4324	0.4314	0.4302
2461	8.37E-06	1.38E+11	-6.03E-03	-5.88E-03	-5.69E-03	-5.48E-03	0.7934	0.7752	0.7523	0.7243	0.4295	0.4288	0.4279	0.4268
2461	8.37E-06	1.75E+11	-5.56E-03	-5.41E-03	-5.24E-03	-5.05E-03	0.7287	0.7121	0.6912	0.6656	0.4270	0.4263	0.4255	0.4246
2461	8.37E-06	2.13E+11	-5.21E-03	-5.07E-03	-4.91E-03	-4.72E-03	0.6809	0.6655	0.6460	0.6222	0.4251	0.4246	0.4238	0.4229
2461	8.37E-06	2.50E+11	-4.93E-03	-4.80E-03	-4.65E-03	-4.47E-03	0.6435	0.6290	0.6106	0.5881	0.4237	0.4232	0.4225	0.4216
2461	8.37E-06	2.88E+11	-4.70E-03	-4.58E-03	-4.43E-03	-4.27E-03	0.6131	0.5993	0.5818	0.5605	0.4226	0.4220	0.4214	0.4206
2461	8.37E-06	3.25E+11	-4.51E-03	-4.39E-03	-4.26E-03	-4.09E-03	0.5877	0.5744	0.5578	0.5373	0.4216	0.4211	0.4205	0.4197
2461	8.37E-06	3.63E+11	-4.35E-03	-4.23E-03	-4.10E-03	-3.95E-03	0.5660	0.5533	0.5372	0.5175	0.4208	0.4203	0.4197	0.4189
2461	8.37E-06	4.00E+11	-4.21E-03	-4.10E-03	-3.97E-03	-3.82E-03	0.5472	0.5349	0.5194	0.5004	0.4201	0.4196	0.4190	0.4183
2461	8.37E-06	4.38E+11	-4.08E-03	-3.98E-03	-3.85E-03	-3.70E-03	0.5306	0.5187	0.5037	0.4853	0.4194	0.4190	0.4184	0.4177
2461	8.37E-06	4.75E+11	-3.97E-03	-3.87E-03	-3.75E-03	-3.60E-03	0.5159	0.5043	0.4897	0.4718	0.4189	0.4184	0.4179	0.4172
2461	8.37E-06	5.13E+11	-3.87E-03	-3.77E-03	-3.65E-03	-3.51E-03	0.5026	0.4914	0.4772	0.4598	0.4184	0.4179	0.4174	0.4168
2461	8.37E-06	5.50E+11	-3.78E-03	-3.68E-03	-3.57E-03	-3.43E-03	0.4907	0.4797	0.4658	0.4488	0.4179	0.4175	0.4170	0.4163
2461	8.37E-06	5.88E+11	-3.70E-03	-3.60E-03	-3.49E-03	-3.35E-03	0.4797	0.4690	0.4555	0.4389	0.4175	0.4171	0.4166	0.4160
2461	8.37E-06	6.25E+11	-3.62E-03	-3.53E-03	-3.42E-03	-3.29E-03	0.4697	0.4592	0.4460	0.4297	0.4171	0.4167	0.4162	0.4156
2461	8.37E-06	6.63E+11	-3.55E-03	-3.46E-03	-3.35E-03	-3.22E-03	0.4605	0.4502	0.4372	0.4213	0.4168	0.4164	0.4159	0.4153
2461	8.37E-06	7.00E+11	-3.49E-03	-3.39E-03	-3.29E-03	-3.16E-03	0.4519	0.4418	0.4291	0.4135	0.4165	0.4161	0.4156	0.4150
2461	8.37E-06	7.38E+11	-3.43E-03	-3.34E-03	-3.23E-03	-3.11E-03	0.4440	0.4341	0.4216	0.4063	0.4162	0.4158	0.4153	0.4148
2461	8.37E-06	7.75E+11	-3.37E-03	-3.28E-03	-3.18E-03	-3.06E-03	0.4366	0.4268	0.4146	0.3995	0.4159	0.4155	0.4151	0.4145
2461	8.37E-06	8.13E+11	-3.32E-03	-3.23E-03	-3.13E-03	-3.01E-03	0.4296	0.4200	0.4080	0.3931	0.4156	0.4153	0.4148	0.4143
2461	8.37E-06	8.50E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4231	0.4137	0.4018	0.3872	0.4154	0.4150	0.4146	0.4140
2461	8.37E-06	8.88E+11	-3.22E-03	-3.14E-03	-3.04E-03	-2.92E-03	0.4169	0.4076	0.3959	0.3815	0.4151	0.4148	0.4144	0.4138
2461	8.37E-06	9.25E+11	-3.18E-03	-3.09E-03	-2.99E-03	-2.88E-03	0.4111	0.4020	0.3904	0.3762	0.4149	0.4146	0.4142	0.4136
2461	8.37E-06	9.63E+11	-3.13E-03	-3.05E-03	-2.96E-03	-2.84E-03	0.4056	0.3966	0.3852	0.3712	0.4147	0.4144	0.4140	0.4134
2461	8.37E-06	1.00E+12	-3.09E-03	-3.01E-03	-2.92E-03	-2.81E-03	0.4003	0.3915	0.3801	0.3663	0.4145	0.4142	0.4138	0.4133
2569	8.02E-06	1.00E+11	-6.82E-03	-6.64E-03	-6.44E-03	-6.19E-03	0.9026	0.8817	0.8551	0.8230	0.4337	0.4329	0.4319	0.4306
2569	8.02E-06	1.38E+11	-6.12E-03	-5.96E-03	-5.77E-03	-5.56E-03	0.8056	0.7871	0.7638	0.7353	0.4299	0.4292	0.4283	0.4272

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2569	8.02E-06	1.75E+11	-5.64E-03	-5.49E-03	-5.32E-03	-5.12E-03	0.7399	0.7229	0.7017	0.6757	0.4274	0.4268	0.4259	0.4250
2569	8.02E-06	2.13E+11	-5.28E-03	-5.14E-03	-4.98E-03	-4.79E-03	0.6912	0.6755	0.6558	0.6315	0.4255	0.4249	0.4242	0.4233
2569	8.02E-06	2.50E+11	-5.00E-03	-4.87E-03	-4.72E-03	-4.54E-03	0.6532	0.6384	0.6198	0.5969	0.4241	0.4235	0.4228	0.4219
2569	8.02E-06	2.88E+11	-4.77E-03	-4.65E-03	-4.50E-03	-4.33E-03	0.6223	0.6082	0.5905	0.5688	0.4229	0.4224	0.4217	0.4209
2569	8.02E-06	3.25E+11	-4.58E-03	-4.46E-03	-4.32E-03	-4.15E-03	0.5965	0.5830	0.5661	0.5453	0.4219	0.4214	0.4208	0.4200
2569	8.02E-06	3.63E+11	-4.41E-03	-4.30E-03	-4.16E-03	-4.00E-03	0.5744	0.5615	0.5452	0.5252	0.4211	0.4206	0.4200	0.4192
2569	8.02E-06	4.00E+11	-4.27E-03	-4.16E-03	-4.03E-03	-3.87E-03	0.5553	0.5428	0.5271	0.5078	0.4204	0.4199	0.4193	0.4186
2569	8.02E-06	4.38E+11	-4.14E-03	-4.03E-03	-3.91E-03	-3.76E-03	0.5385	0.5264	0.5112	0.4925	0.4197	0.4193	0.4187	0.4180
2569	8.02E-06	4.75E+11	-4.03E-03	-3.92E-03	-3.80E-03	-3.66E-03	0.5235	0.5118	0.4970	0.4788	0.4192	0.4187	0.4182	0.4175
2569	8.02E-06	5.13E+11	-3.93E-03	-3.82E-03	-3.70E-03	-3.56E-03	0.5101	0.4986	0.4842	0.4666	0.4186	0.4182	0.4177	0.4170
2569	8.02E-06	5.50E+11	-3.83E-03	-3.73E-03	-3.62E-03	-3.48E-03	0.4979	0.4868	0.4727	0.4555	0.4182	0.4178	0.4172	0.4166
2569	8.02E-06	5.88E+11	-3.75E-03	-3.65E-03	-3.54E-03	-3.40E-03	0.4868	0.4759	0.4622	0.4453	0.4178	0.4174	0.4168	0.4162
2569	8.02E-06	6.25E+11	-3.67E-03	-3.58E-03	-3.46E-03	-3.33E-03	0.4766	0.4660	0.4526	0.4361	0.4174	0.4170	0.4165	0.4159
2569	8.02E-06	6.63E+11	-3.60E-03	-3.51E-03	-3.40E-03	-3.27E-03	0.4673	0.4568	0.4437	0.4275	0.4170	0.4166	0.4162	0.4155
2569	8.02E-06	7.00E+11	-3.54E-03	-3.44E-03	-3.34E-03	-3.21E-03	0.4586	0.4483	0.4354	0.4196	0.4167	0.4163	0.4158	0.4153
2569	8.02E-06	7.38E+11	-3.48E-03	-3.38E-03	-3.28E-03	-3.15E-03	0.4505	0.4405	0.4278	0.4122	0.4164	0.4160	0.4156	0.4150
2569	8.02E-06	7.75E+11	-3.42E-03	-3.33E-03	-3.22E-03	-3.10E-03	0.4430	0.4331	0.4206	0.4053	0.4161	0.4158	0.4153	0.4147
2569	8.02E-06	8.13E+11	-3.36E-03	-3.28E-03	-3.17E-03	-3.05E-03	0.4359	0.4262	0.4140	0.3989	0.4159	0.4155	0.4150	0.4145
2569	8.02E-06	8.50E+11	-3.31E-03	-3.23E-03	-3.13E-03	-3.01E-03	0.4293	0.4197	0.4077	0.3928	0.4156	0.4153	0.4148	0.4143
2569	8.02E-06	8.88E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4231	0.4136	0.4017	0.3871	0.4154	0.4150	0.4146	0.4140
2569	8.02E-06	9.25E+11	-3.22E-03	-3.14E-03	-3.04E-03	-2.92E-03	0.4172	0.4079	0.3962	0.3817	0.4152	0.4148	0.4144	0.4138
2569	8.02E-06	9.63E+11	-3.18E-03	-3.10E-03	-3.00E-03	-2.88E-03	0.4116	0.4024	0.3909	0.3766	0.4149	0.4146	0.4142	0.4136
2569	8.02E-06	1.00E+12	-3.14E-03	-3.06E-03	-2.96E-03	-2.85E-03	0.4063	0.3972	0.3858	0.3718	0.4148	0.4144	0.4140	0.4135
2677	7.69E-06	1.00E+11	-6.92E-03	-6.74E-03	-6.53E-03	-6.28E-03	0.9160	0.8948	0.8681	0.8352	0.4342	0.4334	0.4324	0.4311
2677	7.69E-06	1.38E+11	-6.21E-03	-6.05E-03	-5.86E-03	-5.63E-03	0.8174	0.7986	0.7750	0.7461	0.4304	0.4297	0.4288	0.4277
2677	7.69E-06	1.75E+11	-5.72E-03	-5.57E-03	-5.40E-03	-5.19E-03	0.7507	0.7335	0.7119	0.6855	0.4278	0.4272	0.4263	0.4253
2677	7.69E-06	2.13E+11	-5.36E-03	-5.22E-03	-5.05E-03	-4.86E-03	0.7012	0.6853	0.6652	0.6406	0.4259	0.4253	0.4245	0.4236
2677	7.69E-06	2.50E+11	-5.07E-03	-4.94E-03	-4.78E-03	-4.60E-03	0.6626	0.6476	0.6287	0.6055	0.4244	0.4239	0.4232	0.4223

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2677	7.69E-06	2.88E+11	-4.84E-03	-4.71E-03	-4.56E-03	-4.39E-03	0.6312	0.6169	0.5990	0.5769	0.4232	0.4227	0.4220	0.4212
2677	7.69E-06	3.25E+11	-4.64E-03	-4.52E-03	-4.38E-03	-4.21E-03	0.6050	0.5914	0.5742	0.5531	0.4222	0.4217	0.4211	0.4203
2677	7.69E-06	3.63E+11	-4.47E-03	-4.36E-03	-4.22E-03	-4.06E-03	0.5826	0.5695	0.5530	0.5327	0.4214	0.4209	0.4203	0.4195
2677	7.69E-06	4.00E+11	-4.33E-03	-4.21E-03	-4.08E-03	-3.93E-03	0.5632	0.5505	0.5346	0.5150	0.4207	0.4202	0.4196	0.4188
2677	7.69E-06	4.38E+11	-4.20E-03	-4.09E-03	-3.96E-03	-3.81E-03	0.5461	0.5339	0.5184	0.4994	0.4200	0.4195	0.4190	0.4183
2677	7.69E-06	4.75E+11	-4.08E-03	-3.98E-03	-3.85E-03	-3.71E-03	0.5310	0.5190	0.5040	0.4856	0.4194	0.4190	0.4184	0.4177
2677	7.69E-06	5.13E+11	-3.98E-03	-3.88E-03	-3.76E-03	-3.61E-03	0.5173	0.5057	0.4911	0.4732	0.4189	0.4185	0.4179	0.4173
2677	7.69E-06	5.50E+11	-3.89E-03	-3.79E-03	-3.67E-03	-3.53E-03	0.5050	0.4936	0.4794	0.4619	0.4185	0.4180	0.4175	0.4168
2677	7.69E-06	5.88E+11	-3.80E-03	-3.70E-03	-3.59E-03	-3.45E-03	0.4937	0.4827	0.4687	0.4516	0.4180	0.4176	0.4171	0.4165
2677	7.69E-06	6.25E+11	-3.73E-03	-3.63E-03	-3.51E-03	-3.38E-03	0.4834	0.4726	0.4590	0.4422	0.4176	0.4172	0.4167	0.4161
2677	7.69E-06	6.63E+11	-3.65E-03	-3.56E-03	-3.45E-03	-3.31E-03	0.4739	0.4633	0.4499	0.4335	0.4173	0.4169	0.4164	0.4158
2677	7.69E-06	7.00E+11	-3.59E-03	-3.49E-03	-3.38E-03	-3.25E-03	0.4651	0.4547	0.4416	0.4255	0.4170	0.4166	0.4161	0.4155
2677	7.69E-06	7.38E+11	-3.52E-03	-3.43E-03	-3.32E-03	-3.20E-03	0.4569	0.4467	0.4338	0.4180	0.4166	0.4163	0.4158	0.4152
2677	7.69E-06	7.75E+11	-3.47E-03	-3.37E-03	-3.27E-03	-3.14E-03	0.4492	0.4392	0.4266	0.4110	0.4164	0.4160	0.4155	0.4149
2677	7.69E-06	8.13E+11	-3.41E-03	-3.32E-03	-3.22E-03	-3.09E-03	0.4421	0.4322	0.4198	0.4045	0.4161	0.4157	0.4153	0.4147
2677	7.69E-06	8.50E+11	-3.36E-03	-3.27E-03	-3.17E-03	-3.05E-03	0.4353	0.4256	0.4134	0.3984	0.4158	0.4155	0.4150	0.4145
2677	7.69E-06	8.88E+11	-3.31E-03	-3.22E-03	-3.12E-03	-3.00E-03	0.4290	0.4194	0.4074	0.3926	0.4156	0.4152	0.4148	0.4142
2677	7.69E-06	9.25E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4230	0.4136	0.4017	0.3871	0.4154	0.4150	0.4146	0.4140
2677	7.69E-06	9.63E+11	-3.22E-03	-3.14E-03	-3.04E-03	-2.92E-03	0.4174	0.4081	0.3963	0.3819	0.4152	0.4148	0.4144	0.4138
2677	7.69E-06	1.00E+12	-3.18E-03	-3.10E-03	-3.00E-03	-2.89E-03	0.4120	0.4028	0.3912	0.3770	0.4150	0.4146	0.4142	0.4137
2784	7.40E-06	1.00E+11	-7.01E-03	-6.83E-03	-6.62E-03	-6.37E-03	0.9291	0.9075	0.8804	0.8472	0.4347	0.4339	0.4328	0.4316
2784	7.40E-06	1.38E+11	-6.29E-03	-6.13E-03	-5.93E-03	-5.71E-03	0.8289	0.8098	0.7858	0.7565	0.4308	0.4301	0.4292	0.4281
2784	7.40E-06	1.75E+11	-5.80E-03	-5.65E-03	-5.47E-03	-5.26E-03	0.7611	0.7437	0.7218	0.6950	0.4282	0.4276	0.4267	0.4257
2784	7.40E-06	2.13E+11	-5.43E-03	-5.29E-03	-5.12E-03	-4.93E-03	0.7109	0.6947	0.6744	0.6494	0.4263	0.4257	0.4249	0.4239
2784	7.40E-06	2.50E+11	-5.14E-03	-5.00E-03	-4.85E-03	-4.66E-03	0.6717	0.6565	0.6373	0.6138	0.4248	0.4242	0.4235	0.4226
2784	7.40E-06	2.88E+11	-4.90E-03	-4.77E-03	-4.62E-03	-4.45E-03	0.6399	0.6254	0.6072	0.5848	0.4236	0.4230	0.4223	0.4215
2784	7.40E-06	3.25E+11	-4.70E-03	-4.58E-03	-4.44E-03	-4.27E-03	0.6133	0.5994	0.5820	0.5606	0.4226	0.4220	0.4214	0.4206
2784	7.40E-06	3.63E+11	-4.53E-03	-4.41E-03	-4.28E-03	-4.11E-03	0.5906	0.5772	0.5605	0.5399	0.4217	0.4212	0.4206	0.4198

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2784	7.40E-06	4.00E+11	-4.39E-03	-4.27E-03	-4.14E-03	-3.98E-03	0.5709	0.5580	0.5418	0.5220	0.4209	0.4205	0.4199	0.4191
2784	7.40E-06	4.38E+11	-4.25E-03	-4.14E-03	-4.01E-03	-3.86E-03	0.5535	0.5411	0.5254	0.5062	0.4203	0.4198	0.4192	0.4185
2784	7.40E-06	4.75E+11	-4.14E-03	-4.03E-03	-3.90E-03	-3.76E-03	0.5381	0.5260	0.5108	0.4921	0.4197	0.4193	0.4187	0.4180
2784	7.40E-06	5.13E+11	-4.03E-03	-3.93E-03	-3.81E-03	-3.66E-03	0.5243	0.5125	0.4977	0.4795	0.4192	0.4187	0.4182	0.4175
2784	7.40E-06	5.50E+11	-3.94E-03	-3.84E-03	-3.72E-03	-3.57E-03	0.5118	0.5003	0.4859	0.4681	0.4187	0.4183	0.4177	0.4171
2784	7.40E-06	5.88E+11	-3.85E-03	-3.75E-03	-3.63E-03	-3.50E-03	0.5004	0.4892	0.4750	0.4577	0.4183	0.4179	0.4173	0.4167
2784	7.40E-06	6.25E+11	-3.77E-03	-3.68E-03	-3.56E-03	-3.42E-03	0.4899	0.4789	0.4651	0.4482	0.4179	0.4175	0.4170	0.4163
2784	7.40E-06	6.63E+11	-3.70E-03	-3.60E-03	-3.49E-03	-3.36E-03	0.4803	0.4695	0.4560	0.4394	0.4175	0.4171	0.4166	0.4160
2784	7.40E-06	7.00E+11	-3.63E-03	-3.54E-03	-3.43E-03	-3.30E-03	0.4713	0.4608	0.4475	0.4312	0.4172	0.4168	0.4163	0.4157
2784	7.40E-06	7.38E+11	-3.57E-03	-3.48E-03	-3.37E-03	-3.24E-03	0.4630	0.4527	0.4396	0.4236	0.4169	0.4165	0.4160	0.4154
2784	7.40E-06	7.75E+11	-3.51E-03	-3.42E-03	-3.31E-03	-3.19E-03	0.4553	0.4451	0.4323	0.4165	0.4166	0.4162	0.4157	0.4151
2784	7.40E-06	8.13E+11	-3.46E-03	-3.37E-03	-3.26E-03	-3.14E-03	0.4480	0.4380	0.4254	0.4099	0.4163	0.4159	0.4155	0.4149
2784	7.40E-06	8.50E+11	-3.40E-03	-3.32E-03	-3.21E-03	-3.09E-03	0.4412	0.4313	0.4189	0.4037	0.4161	0.4157	0.4152	0.4147
2784	7.40E-06	8.88E+11	-3.36E-03	-3.27E-03	-3.16E-03	-3.04E-03	0.4348	0.4251	0.4128	0.3978	0.4158	0.4155	0.4150	0.4144
2784	7.40E-06	9.25E+11	-3.31E-03	-3.22E-03	-3.12E-03	-3.00E-03	0.4287	0.4191	0.4071	0.3923	0.4156	0.4152	0.4148	0.4142
2784	7.40E-06	9.63E+11	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4229	0.4135	0.4016	0.3870	0.4154	0.4150	0.4146	0.4140
2784	7.40E-06	1.00E+12	-3.22E-03	-3.14E-03	-3.04E-03	-2.93E-03	0.4175	0.4082	0.3965	0.3821	0.4152	0.4148	0.4144	0.4139
2892	7.12E-06	1.00E+11	-7.10E-03	-6.92E-03	-6.70E-03	-6.45E-03	0.9420	0.9200	0.8925	0.8588	0.4352	0.4344	0.4333	0.4320
2892	7.12E-06	1.38E+11	-6.37E-03	-6.21E-03	-6.01E-03	-5.78E-03	0.8402	0.8208	0.7965	0.7667	0.4313	0.4305	0.4296	0.4285
2892	7.12E-06	1.75E+11	-5.87E-03	-5.72E-03	-5.54E-03	-5.33E-03	0.7713	0.7537	0.7315	0.7043	0.4286	0.4279	0.4271	0.4261
2892	7.12E-06	2.13E+11	-5.50E-03	-5.35E-03	-5.19E-03	-4.99E-03	0.7202	0.7040	0.6834	0.6581	0.4267	0.4260	0.4252	0.4243
2892	7.12E-06	2.50E+11	-5.20E-03	-5.07E-03	-4.91E-03	-4.72E-03	0.6807	0.6652	0.6458	0.6219	0.4251	0.4245	0.4238	0.4229
2892	7.12E-06	2.88E+11	-4.96E-03	-4.83E-03	-4.68E-03	-4.51E-03	0.6484	0.6337	0.6152	0.5925	0.4239	0.4233	0.4226	0.4218
2892	7.12E-06	3.25E+11	-4.76E-03	-4.64E-03	-4.49E-03	-4.32E-03	0.6214	0.6073	0.5897	0.5680	0.4229	0.4223	0.4217	0.4208
2892	7.12E-06	3.63E+11	-4.59E-03	-4.47E-03	-4.33E-03	-4.17E-03	0.5984	0.5849	0.5679	0.5470	0.4220	0.4215	0.4208	0.4200
2892	7.12E-06	4.00E+11	-4.44E-03	-4.33E-03	-4.19E-03	-4.03E-03	0.5784	0.5654	0.5490	0.5288	0.4212	0.4207	0.4201	0.4194
2892	7.12E-06	4.38E+11	-4.31E-03	-4.20E-03	-4.06E-03	-3.91E-03	0.5608	0.5482	0.5323	0.5128	0.4206	0.4201	0.4195	0.4188
2892	7.12E-06	4.75E+11	-4.19E-03	-4.08E-03	-3.95E-03	-3.80E-03	0.5452	0.5330	0.5175	0.4986	0.4200	0.4195	0.4189	0.4182

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
2892	7.12E-06	5.13E+11	-4.09E-03	-3.98E-03	-3.85E-03	-3.71E-03	0.5312	0.5193	0.5042	0.4858	0.4194	0.4190	0.4184	0.4177
2892	7.12E-06	5.50E+11	-3.99E-03	-3.89E-03	-3.76E-03	-3.62E-03	0.5185	0.5069	0.4922	0.4742	0.4190	0.4185	0.4180	0.4173
2892	7.12E-06	5.88E+11	-3.90E-03	-3.80E-03	-3.68E-03	-3.54E-03	0.5069	0.4956	0.4813	0.4637	0.4185	0.4181	0.4176	0.4169
2892	7.12E-06	6.25E+11	-3.82E-03	-3.72E-03	-3.61E-03	-3.47E-03	0.4963	0.4852	0.4712	0.4540	0.4181	0.4177	0.4172	0.4165
2892	7.12E-06	6.63E+11	-3.75E-03	-3.65E-03	-3.54E-03	-3.40E-03	0.4865	0.4756	0.4619	0.4451	0.4178	0.4174	0.4168	0.4162
2892	7.12E-06	7.00E+11	-3.68E-03	-3.58E-03	-3.47E-03	-3.34E-03	0.4775	0.4668	0.4533	0.4368	0.4174	0.4170	0.4165	0.4159
2892	7.12E-06	7.38E+11	-3.62E-03	-3.52E-03	-3.41E-03	-3.28E-03	0.4691	0.4586	0.4454	0.4291	0.4171	0.4167	0.4162	0.4156
2892	7.12E-06	7.75E+11	-3.56E-03	-3.46E-03	-3.35E-03	-3.23E-03	0.4612	0.4509	0.4379	0.4220	0.4168	0.4164	0.4159	0.4153
2892	7.12E-06	8.13E+11	-3.50E-03	-3.41E-03	-3.30E-03	-3.18E-03	0.4538	0.4437	0.4309	0.4152	0.4165	0.4162	0.4157	0.4151
2892	7.12E-06	8.50E+11	-3.45E-03	-3.36E-03	-3.25E-03	-3.13E-03	0.4469	0.4370	0.4244	0.4089	0.4163	0.4159	0.4154	0.4149
2892	7.12E-06	8.88E+11	-3.40E-03	-3.31E-03	-3.21E-03	-3.08E-03	0.4404	0.4306	0.4182	0.4030	0.4160	0.4157	0.4152	0.4146
2892	7.12E-06	9.25E+11	-3.35E-03	-3.26E-03	-3.16E-03	-3.04E-03	0.4343	0.4246	0.4124	0.3974	0.4158	0.4154	0.4150	0.4144
2892	7.12E-06	9.63E+11	-3.31E-03	-3.22E-03	-3.12E-03	-3.00E-03	0.4284	0.4189	0.4069	0.3921	0.4156	0.4152	0.4148	0.4142
2892	7.12E-06	1.00E+12	-3.27E-03	-3.18E-03	-3.08E-03	-2.96E-03	0.4229	0.4135	0.4016	0.3870	0.4154	0.4150	0.4146	0.4140
3000	6.86E-06	1.00E+11	-7.19E-03	-7.01E-03	-6.79E-03	-6.53E-03	0.9546	0.9323	0.9044	0.8702	0.4357	0.4349	0.4338	0.4325
3000	6.86E-06	1.38E+11	-6.45E-03	-6.28E-03	-6.09E-03	-5.86E-03	0.8512	0.8316	0.8069	0.7768	0.4317	0.4310	0.4300	0.4288
3000	6.86E-06	1.75E+11	-5.95E-03	-5.79E-03	-5.61E-03	-5.40E-03	0.7814	0.7635	0.7410	0.7134	0.4290	0.4283	0.4275	0.4264
3000	6.86E-06	2.13E+11	-5.57E-03	-5.42E-03	-5.25E-03	-5.05E-03	0.7298	0.7131	0.6922	0.6666	0.4270	0.4264	0.4256	0.4246
3000	6.86E-06	2.50E+11	-5.27E-03	-5.13E-03	-4.97E-03	-4.78E-03	0.6895	0.6738	0.6541	0.6299	0.4255	0.4249	0.4241	0.4232
3000	6.86E-06	2.88E+11	-5.03E-03	-4.89E-03	-4.74E-03	-4.56E-03	0.6567	0.6418	0.6231	0.6001	0.4242	0.4237	0.4229	0.4221
3000	6.86E-06	3.25E+11	-4.82E-03	-4.70E-03	-4.55E-03	-4.38E-03	0.6293	0.6151	0.5972	0.5752	0.4232	0.4226	0.4220	0.4211
3000	6.86E-06	3.63E+11	-4.65E-03	-4.53E-03	-4.38E-03	-4.22E-03	0.6060	0.5923	0.5751	0.5540	0.4223	0.4218	0.4211	0.4203
3000	6.86E-06	4.00E+11	-4.50E-03	-4.38E-03	-4.24E-03	-4.08E-03	0.5858	0.5725	0.5559	0.5355	0.4215	0.4210	0.4204	0.4196
3000	6.86E-06	4.38E+11	-4.36E-03	-4.25E-03	-4.12E-03	-3.96E-03	0.5680	0.5552	0.5391	0.5193	0.4208	0.4204	0.4197	0.4190
3000	6.86E-06	4.75E+11	-4.24E-03	-4.13E-03	-4.00E-03	-3.85E-03	0.5521	0.5397	0.5241	0.5049	0.4202	0.4198	0.4192	0.4185
3000	6.86E-06	5.13E+11	-4.14E-03	-4.03E-03	-3.90E-03	-3.75E-03	0.5379	0.5258	0.5106	0.4919	0.4197	0.4192	0.4187	0.4180
3000	6.86E-06	5.50E+11	-4.04E-03	-3.93E-03	-3.81E-03	-3.67E-03	0.5250	0.5133	0.4984	0.4802	0.4192	0.4188	0.4182	0.4175
3000	6.86E-06	5.88E+11	-3.95E-03	-3.85E-03	-3.73E-03	-3.59E-03	0.5133	0.5018	0.4873	0.4695	0.4188	0.4183	0.4178	0.4171

Continued on next page

Table 14 – continued from previous page

$\rho\ (kg/m^3)$	$t\ (m)$	$E\ (Pa)$	$U_3\ (m)$				$SE\ (J)$				W/B_{skin}			
			$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
3000	6.86E-06	6.25E+11	-3.87E-03	-3.77E-03	-3.65E-03	-3.51E-03	0.5026	0.4913	0.4771	0.4597	0.4184	0.4179	0.4174	0.4168
3000	6.86E-06	6.63E+11	-3.80E-03	-3.70E-03	-3.58E-03	-3.44E-03	0.4927	0.4816	0.4677	0.4507	0.4180	0.4176	0.4171	0.4164
3000	6.86E-06	7.00E+11	-3.73E-03	-3.63E-03	-3.51E-03	-3.38E-03	0.4835	0.4727	0.4590	0.4423	0.4176	0.4172	0.4167	0.4161
3000	6.86E-06	7.38E+11	-3.66E-03	-3.57E-03	-3.45E-03	-3.32E-03	0.4750	0.4643	0.4510	0.4345	0.4173	0.4169	0.4164	0.4158
3000	6.86E-06	7.75E+11	-3.60E-03	-3.51E-03	-3.40E-03	-3.27E-03	0.4670	0.4566	0.4434	0.4273	0.4170	0.4166	0.4161	0.4155
3000	6.86E-06	8.13E+11	-3.54E-03	-3.45E-03	-3.34E-03	-3.22E-03	0.4595	0.4493	0.4363	0.4204	0.4167	0.4164	0.4159	0.4153
3000	6.86E-06	8.50E+11	-3.49E-03	-3.40E-03	-3.29E-03	-3.17E-03	0.4525	0.4424	0.4297	0.4141	0.4165	0.4161	0.4156	0.4150
3000	6.86E-06	8.88E+11	-3.44E-03	-3.35E-03	-3.24E-03	-3.12E-03	0.4459	0.4360	0.4235	0.4080	0.4162	0.4159	0.4154	0.4148
3000	6.86E-06	9.25E+11	-3.39E-03	-3.30E-03	-3.20E-03	-3.08E-03	0.4397	0.4299	0.4175	0.4023	0.4160	0.4156	0.4152	0.4146
3000	6.86E-06	9.63E+11	-3.35E-03	-3.26E-03	-3.16E-03	-3.04E-03	0.4338	0.4241	0.4120	0.3970	0.4158	0.4154	0.4150	0.4144
3000	6.86E-06	1.00E+12	-3.31E-03	-3.22E-03	-3.12E-03	-3.00E-03	0.4282	0.4187	0.4066	0.3919	0.4156	0.4152	0.4148	0.4142

A.7 Frame Standalone Model Convergence Study

The following table shows the results from the frame standalone mode convergence study, using B32 beam elements. The model properties are: modulus of elasticity (303 GPa), Poison's ratio (0.18) and an icosahedron radius of 0.1524 m. A linear buckling analysis is performed considering the first five critical pressures (P_{cr}). **Edge Seed** refers to the amount of elements along each edge and **# Elements** is the resulting number of elements. $\boldsymbol{P_{cr,MD}: i \,\&\, i-1}$ is the maximum error of all P_{cr} comparing row i and row $i-1$. $\boldsymbol{P_{cr,MD}: i \,\&\, i_{end}}$ is the maximum error of all P_{cr} comparing row i and row i_{end}.

Table 15: Frame Standalone Model Convergence Study Results

Edge Seed	# Elements	P_{cr1} (Pa)	P_{cr2} (Pa)	P_{cr3} (Pa)	P_{cr4} (Pa)	P_{cr5} (Pa)	$P_{cr,MD}: i \,\&\, i-1$	$P_{cr,MD}: i \,\&\, i_{end}$
5	150	5.22E-13	1.95E-01	1.95E-01	1.95E-01	2.31E-01	-	200%
6	180	3.88E-13	1.95E-01	1.95E-01	1.95E-01	2.30E-01	29.399%	200%
7	210	1.25E-12	1.95E-01	1.95E-01	1.95E-01	2.30E-01	105.318%	200%
8	240	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	200.000%	0.022%
9	270	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.010%	0.013%
10	300	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.005%	0.010%
11	330	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.005%	0.005%
12	360	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.005%
13	390	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.005%	0.005%
14	420	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.005%
15	450	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.004%	0.005%
16	480	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.005%	0.005%
17	510	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.005%
18	540	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.005%
19	570	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.005%
20	600	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.005%
21	630	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.005%	0.000%
22	660	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.000%
23	690	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.000%
24	720	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.000%
25	750	1.18E-02	1.95E-01	1.95E-01	2.07E-01	2.30E-01	0.000%	0.000%

A.8 Frame Standalone Beam Profile Study

The following table shows the results from the frame beam profile study. The model properties are: modulus of elasticity (303 GPa), Poison's ratio (0.18), 600 B32 elements, icosahedron radius (0.1524 m) and frame W/B of 0.35 (see Equation (2.18)) with circular beam profile. Both linear static and buckling analyses are performed, comparing the radius (r_n), moment of inertia (I_n), maximum displacement ($U_{max,n}$), maximum von Mises stress ($S_{max,n}$) and critical pressure ($P_{cr,n}$) change with beam thickness to radius ratio(c). All results are normalized against the solid beam profile.

Table 16: Frame Beam Profile Study Results

c	r_n	I_n	$U_{max,n}$	$S_{max,n}$	$P_{crit1,n}$	$P_{crit2,n}$
0.05	3.20	19.51	0.05	0.17	19.46	19.42
0.1	2.29	9.53	0.11	0.24	9.49	9.48
0.15	1.90	6.21	0.16	0.29	6.16	6.16
0.2	1.67	4.56	0.22	0.34	4.50	4.50
0.25	1.51	3.57	0.29	0.38	3.50	3.50
0.3	1.40	2.92	0.35	0.43	2.83	2.83
0.35	1.32	2.46	0.43	0.47	2.36	2.36
0.4	1.25	2.13	0.50	0.50	2.00	2.00
0.45	1.20	1.87	0.58	0.54	1.72	1.72
0.5	1.15	1.67	0.67	0.58	1.50	1.50
0.55	1.12	1.51	0.76	0.62	1.32	1.32
0.6	1.09	1.38	0.86	0.66	1.17	1.17
0.65	1.07	1.28	0.96	0.70	1.04	1.04
0.7	1.05	1.20	1.08	0.74	0.93	0.93
0.75	1.03	1.13	1.20	0.78	0.83	0.83
0.8	1.02	1.08	1.33	0.82	0.75	0.75
0.85	1.01	1.05	1.48	0.86	0.68	0.68

Appendix B: Python Codes

B.1 Circular Model

```
# -*- coding: mbcs -*-
from part import*
from material import*
from section import*
from assembly import*
from step import*
from interaction import*
from load import*
from mesh import*
from job import*
from sketch import*
from visualization import*
from connectorBehavior import*
from odbAccess import *
from abaqusConstants import *
from odbMaterial import *
from odbSection import *
import os
session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

# Sets Working Directory ******************************************************************************
os.chdir(path)

# Load variables **************************************************************************************
execfile('Var_circle.py')

# Model/Job name and creation *************************************************************************
mdb.Model(modelType=STANDARD_EXPLICIT,name=model_name)

# Creates Circular Membrane ***************************************************************************
mdb.models[model_name].ConstrainedSketch(name='profile',sheetSize=200.0)
mdb.models[model_name].sketches['profile'].CircleByCenterPerimeter(
    center=(0.0, 0.0), point1=(0.0,r_circular_membrane))
mdb.models[model_name].Part(dimensionality=THREE_D, name='Circle',type=DEFORMABLE_BODY)
mdb.models[model_name].parts['Circle'].BaseShell(sketch=mdb.models[model_name].sketches['profile'])
skin = mdb.models[model_name].parts['Circle']

# Materials *******************************************************************************
mdb.models[model_name].Material(name='Skin Material')
mdb.models[model_name].materials['Skin Material'].Density(table=((skin_density,),))
mdb.models[model_name].materials['Skin Material'].Elastic(table=((skin_modulus,skin_poisson),))

# Profiles & Section Assignments **********************************************************
if membrane == 1:
    mdb.models[model_name].MembraneSection(material=
        'Skin Material', name='Skin Section', poisson=skin_poisson,poissonDefinition=VALUE,
        thickness=skin_thickness,thicknessField='',thicknessType=UNIFORM)
else:
    mdb.models[model_name].HomogeneousShellSection(idealization=skin_section_idealization,
        material='Skin Material',name='Skin Section',poisson=skin_poisson,poissonDefinition=VALUE,
preIntegrate=ON,thickness=skin_thickness,thicknessField='',thicknessModulus=None,
```

```
        thicknessType=UNIFORM, useDensity=ON)

skin.SectionAssignment(offset=0.0,offsetField='',offsetType=skin_section_location,region=Region(
        faces=skin.faces.findAt(((0,0,0),),)),sectionName='Skin Section',thicknessAssignment=FROM_SECTION)

# Assembly ********************************************************************************
mdb.models[model_name].rootAssembly.Instance(dependent=OFF,name='Skin Instance',part=skin)
skin_instance = mdb.models[model_name].rootAssembly.instances['Skin Instance']
root_assembly = mdb.models[model_name].rootAssembly

# Spherical Datum Convertion ******************************************************************
datumid = skin.DatumCsysByThreePoints(coordSysType=SPHERICAL,
          line1=(1.0, 0.0, 0.0), line2=(0.0, 1.0, 0.0), name='Spherical',origin=(0.0, 0.0, 0.0)).id
skin.MaterialOrientation(
    additionalRotationField='',additionalRotationType=ROTATION_NONE,angle=0.0,axis=AXIS_3, fieldName='',
    localCsys=skin.datums[datumid],
    orientationType=SYSTEM,region=Region(faces=skin.faces.findAt(((0,0,0),))))
mdb.models[model_name].rootAssembly.regenerate()

#Step ***********************************************************************************
if stabilization == 1:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,previous='Initial',
    continueDampingFactors=False,adaptiveDampingRatio=stabilization_ratio,
    stabilizationMagnitude=stabilization_magn,stabilizationMethod=DISSIPATED_ENERGY_FRACTION)
elif step_type == 1:
    mdb.models[model_name].StaticRiksStep(initialArcInc=initial_ArcInc,maxLPF=max_LPF,
    minArcInc=min_ArcInc,maxArcInc=max_ArcInc,maxNumInc=maxnuminc,name=stepname,
    nlgeom=nonlinear_effects,previous='Initial')
else:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,previous='Initial')

#Pressure Load ***************************************************************************
mdb.models[model_name].Pressure(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,field='',magnitude=P,name='Pressure Towards Center',
    region=Region(side1Faces=skin_instance.faces.findAt(((0,0,0),),)))

# BC *************************************************************************************
datumid = root_assembly.DatumCsysByThreePoints(coordSysType=SPHERICAL,
          line1=(1.0, 0.0, 0.0), line2=(0.0, 1.0, 0.0), name='Spherical',origin=(0.0, 0.0, 0.0)).id

if bc_type == 1:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,fieldName='',fixed=OFF,
    localCsys=root_assembly.datums[datumid],name='BC-rotations allowed',
    region=Region(edges=skin_instance.edges.findAt(((r_circular_membrane,0,0),))),
    u1=0.0,u2=0.0,u3=0.0) ## rotations allowed
else:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,fieldName='',fixed=OFF,
    localCsys=root_assembly.datums[datumid],name='BC-encastre',
    region=Region(edges=skin_instance.edges.findAt(((r_circular_membrane,0,0),))),
    u1=0.0,u2=0.0,u3=0.0,ur1=0.0,ur2=0.0,ur3=0.0) ## encastre

# Mesh ***********************************************************************************

# Seeding
```

```
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER,edges=
    skin_instance.edges.findAt(((r_circular_membrane,0,0),)),number=skin_seed_number) # seeding by edge
##root_assembly.seedPartInstance(deviationFactor=0.1, minSizeFactor=0.1,
##    regions=(skin_instance,),size=skin_seed_number) # seeding by instance

# Element Shape
root_assembly.setMeshControls(elemShape=skin_element_shape,regions=
    skin_instance.faces.findAt(((0,0,0),),))
# Element Type
mdb.models[model_name].rootAssembly.setElementType(elemTypes=(ElemType(elemCode=skin_element_type1,
    elemLibrary=STANDARD),ElemType(elemCode=skin_element_type2,elemLibrary=STANDARD,
    secondOrderAccuracy=OFF)),regions=(skin_instance.faces.findAt(((0,0,0),)),))
#Mesh Instance
mdb.models[model_name].rootAssembly.generateMesh(regions=(skin_instance,))

# Field Outputs ***************************************************************************************
if membrane == 1:
    mdb.models[model_name].FieldOutputRequest(name='Field-Output',createStepName=stepname,
                                              variables=('S','U'))
else:
    mdb.models[model_name].FieldOutputRequest(name='Field-Output',createStepName=stepname,
                                              variables=('S','U'),sectionPoints=(1,2,3))

del mdb.models[model_name].fieldOutputRequests['F-Output-1']

# Job *************************************************************************************************
mdb.Job(atTime=None, contactPrint=OFF, description='',echoPrint=OFF,explicitPrecision=SINGLE,
    getMemoryFromAnalysis=False,historyPrint=OFF,memory=memory_usage,memoryUnits=MEGA_BYTES,
    model=model_name,modelPrint=OFF,multiprocessingMode=DEFAULT,name=job_name,nodalOutputPrecision=FULL,
    numCpus=num_cores,numDomains=num_cores,numGPUs=num_GPUs,queue=None,scratch='',type=ANALYSIS,
    userSubroutine='',waitHours=0, waitMinutes=0)

# Submission ******************************************************************************************
mdb.jobs[job_name].submit(consistencyChecking=OFF)
mdb.jobs[job_name].waitForCompletion()
```

B.2 Square Model

```
# -*- coding: mbcs -*-
from part import*
from material import*
from section import*
from assembly import*
from step import*
from interaction import*
from load import*
from mesh import*
from job import*
from sketch import*
from visualization import*
from connectorBehavior import*
from odbAccess import *
from abaqusConstants import *
from odbMaterial import *
from odbSection import *
```

```
import os
import math
session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

# Sets Working Directory ********************************************************************************
os.chdir(path)

# Load variables ****************************************************************************************
execfile('Var_rectangle.py')

# Model/Job name and creation ***************************************************************************
mdb.Model(modelType=STANDARD_EXPLICIT,name=model_name)

# Creates Rectangular Membrane **************************************************************************
mdb.models[model_name].ConstrainedSketch(name='profile',sheetSize=200.0)
mdb.models[model_name].sketches['profile'].rectangle(
    point1=(-rect_width/2, rect_height/2), point2=(rect_width/2, -rect_height/2))
mdb.models[model_name].Part(dimensionality=THREE_D, name='Rectangle',type=DEFORMABLE_BODY)
mdb.models[model_name].parts['Rectangle'].BaseShell(sketch=mdb.models[model_name].sketches['profile'])
skin = mdb.models[model_name].parts['Rectangle']

# Materials *********************************************************************************************
mdb.models[model_name].Material(name='Skin Material')
mdb.models[model_name].materials['Skin Material'].Density(table=((skin_density,),))
mdb.models[model_name].materials['Skin Material'].Elastic(table=((skin_modulus,skin_poisson),))

# Profiles & Section Assignments ************************************************************************
if membrane == 1:
    mdb.models[model_name].MembraneSection(material=
        'Skin Material', name='Skin Section', poisson=skin_poisson,poissonDefinition=VALUE,
        thickness=skin_thickness,thicknessField='',thicknessType=UNIFORM)
else:
    mdb.models[model_name].HomogeneousShellSection(idealization=skin_section_idealization,
        material='Skin Material',name='Skin Section',poisson=skin_poisson,poissonDefinition=VALUE,
preIntegrate=ON,thickness=skin_thickness,thicknessField='',thicknessModulus=None,
        thicknessType=UNIFORM, useDensity=ON)

skin.SectionAssignment(offset=0.0,offsetField='',offsetType=skin_section_location,region=Region(
        faces=skin.faces.findAt(((0,0,0),),)),sectionName='Skin Section',thicknessAssignment=FROM_SECTION)

# Assembly **********************************************************************************************
mdb.models[model_name].rootAssembly.Instance(dependent=OFF,name='Skin Instance',part=skin)
skin_instance = mdb.models[model_name].rootAssembly.instances['Skin Instance']
root_assembly = mdb.models[model_name].rootAssembly

#Step ***************************************************************************************************
if stabilization == 1:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,previous='Initial',
    continueDampingFactors=False,adaptiveDampingRatio=stabilization_ratio,
    stabilizationMagnitude=stabilization_magn,stabilizationMethod=DISSIPATED_ENERGY_FRACTION,
    timeIncrementationMethod=increment_method)
elif step_type == 1:
    mdb.models[model_name].StaticRiksStep(initialArcInc=initial_ArcInc,maxLPF=max_LPF,
    minArcInc=min_ArcInc,maxArcInc=max_ArcInc,maxNumInc=maxnuminc,name=stepname,
    nlgeom=nonlinear_effects,timeIncrementationMethod=increment_method,previous='Initial')
else:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
```

```
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,
    timeIncrementationMethod=increment_method,previous='Initial')

#Pressure Load ********************************************************************************
mdb.models[model_name].Pressure(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,field='',magnitude=P,name='Pressure Towards Center',
    region=Region(side1Faces=skin_instance.faces.findAt(((0,0,0),),)))

# BC ***************************************************************************************
if bc_type == 1:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BC-rotations allowed',
    region=Region(edges=skin_instance.edges.findAt(((rect_width/2,0,0),),((-rect_width/2,0,0),),
    ((0,rect_height/2,0),),((0,-rect_height/2,0),))),u1=0.0,u2=0.0,u3=0.0) ## rotations allowed
else:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BC-rotations allowed',
    region=Region(edges=skin_instance.edges.findAt(((rect_width/2,0,0),),((-rect_width/2,0,0),),
    ((0,rect_height/2,0),),((0,-rect_height/2,0),))),u1=0.0,u2=0.0,u3=0.0,ur1=0.0,ur2=0.0,ur3=0.0) ## encastre

# Mesh *************************************************************************************

# Seeding
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER,edges=
    skin_instance.edges.findAt(((rect_width/2,0,0),),((-rect_width/2,0,0),),
    ((0,rect_height/2,0),),((0,-rect_height/2,0),)),number=skin_seed_number) # seeding by edge
##root_assembly.seedPartInstance(deviationFactor=0.1, minSizeFactor=0.1,
##    regions=(skin_instance,),size=skin_seed_number) # seeding by instance

# Element Shape
root_assembly.setMeshControls(elemShape=skin_element_shape,regions=
    skin_instance.faces.findAt(((0,0,0),),))
# Element Type
mdb.models[model_name].rootAssembly.setElementType(elemTypes=(ElemType(elemCode=skin_element_type1,
    elemLibrary=STANDARD),ElemType(elemCode=skin_element_type2,elemLibrary=STANDARD,
    secondOrderAccuracy=OFF)),regions=(skin_instance.faces.findAt(((0,0,0),)),))
#Mesh Instance
mdb.models[model_name].rootAssembly.generateMesh(regions=(skin_instance,))

# Field Outputs ****************************************************************************************
if membrane == 1:
    mdb.models[model_name].FieldOutputRequest(name='Field-Output',createStepName=stepname,
                                              variables=('S','U'))
else:
    mdb.models[model_name].FieldOutputRequest(name='Field-Output',createStepName=stepname,
                                              variables=('S','U'),sectionPoints=(1,2,3))

del mdb.models[model_name].fieldOutputRequests['F-Output-1']

# Job **************************************************************************************************
mdb.Job(atTime=None, contactPrint=OFF, description='',echoPrint=OFF,explicitPrecision=SINGLE,
    getMemoryFromAnalysis=False,historyPrint=OFF,memory=memory_usage,memoryUnits=MEGA_BYTES,
    model=model_name,modelPrint=OFF,multiprocessingMode=DEFAULT,name=job_name,nodalOutputPrecision=FULL,
    numCpus=num_cores,numDomains=num_cores,numGPUs=num_GPUs,queue=None,scratch='',type=ANALYSIS,
    userSubroutine='',waitHours=0, waitMinutes=0)

# Submission *******************************************************************************************
mdb.jobs[job_name].submit(consistencyChecking=OFF)
```

```
mdb.jobs[job_name].waitForCompletion()
```

B.3 Triangular Model

```
# -*- coding: mbcs -*-
from part import*
from material import*
from section import*
from assembly import*
from step import*
from interaction import*
from load import*
from mesh import*
from job import*
from sketch import*
from visualization import*
from connectorBehavior import*
from odbAccess import *
from abaqusConstants import *
from odbMaterial import *
from odbSection import *
import os
session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

# Sets Working Directory ******************************************************************************
os.chdir(path)

# Load variables **************************************************************************************
execfile('Var_triangle.py')

# Model/Job name and creation *************************************************************************
mdb.Model(modelType=STANDARD_EXPLICIT,name=model_name)

# Creates Skin ****************************************************************************
mdb.models[model_name].Part(dimensionality=THREE_D, name='Skin',type=DEFORMABLE_BODY)
skin=mdb.models[model_name].parts['Skin']
mdb.models[model_name].ConstrainedSketch(name='__profile__', sheetSize=2.0)
mdb.models[model_name].sketches['__profile__']

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p1,p2),(p1,p3),(p2,p3)))
e12=skin.edges.findAt(mp12); e13=skin.edges.findAt(mp13); e23=skin.edges.findAt(mp23)
skin.CoverEdges((e12,e13,e23),tryAnalytical=True)

# Materials *******************************************************************************
mdb.models[model_name].Material(name='Skin Material')
mdb.models[model_name].materials['Skin Material'].Density(table=((skin_density,),))
mdb.models[model_name].materials['Skin Material'].Elastic(table=((skin_modulus,skin_poisson),))

# Profiles & Section Assignments **********************************************************
if membrane == 1:
    mdb.models[model_name].MembraneSection(material=
        'Skin Material', name='Skin Section', poisson=skin_poisson,poissonDefinition=VALUE,
        thickness=skin_thickness,thicknessField='',thicknessType=UNIFORM)
else:
    mdb.models[model_name].HomogeneousShellSection(idealization=skin_section_idealization,
```

```
        material='Skin Material',name='Skin Section',poisson=skin_poisson,poissonDefinition=VALUE,
preIntegrate=ON,thickness=skin_thickness,thicknessField='',thicknessModulus=None,
        thicknessType=UNIFORM, useDensity=ON)

skin.SectionAssignment(offset=0.0,offsetField='',offsetType=skin_section_location,region=Region(
        faces=skin.faces.findAt((fc123,),)),sectionName='Skin Section',thicknessAssignment=FROM_SECTION)

# Assembly ***************************************************************************
mdb.models[model_name].rootAssembly.Instance(dependent=OFF,name='Skin Instance',part=skin)
skin_instance = mdb.models[model_name].rootAssembly.instances['Skin Instance']
root_assembly = mdb.models[model_name].rootAssembly

#Step ********************************************************************************
if stabilization == 1:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,previous='Initial',
    continueDampingFactors=False,adaptiveDampingRatio=stabilization_ratio,
    stabilizationMagnitude=stabilization_magn,stabilizationMethod=DISSIPATED_ENERGY_FRACTION)
elif step_type == 1:
    mdb.models[model_name].StaticRiksStep(initialArcInc=initial_ArcInc,maxLPF=max_LPF,
    minArcInc=min_ArcInc,maxArcInc=max_ArcInc,maxNumInc=maxnuminc,name=stepname,
    nlgeom=nonlinear_effects,previous='Initial')
else:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,previous='Initial')

#Pressure Load ***********************************************************************
mdb.models[model_name].Pressure(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,field='',magnitude=P,name='Pressure Towards Center',
    region=Region(side2Faces=skin_instance.faces.findAt((fc123,),)))

# BC *********************************************************************************
if bc_type == 1:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BC-rotations allowed',
    region=Region(edges=skin_instance.edges.findAt((mp12,),(mp13,),(mp23,),)),
    u1=0.0,u2=0.0,u3=0.0) ## rotations allowed
else:
    mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=stepname,
    distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BC-encastre',
    region=Region(edges=skin_instance.edges.findAt((mp12,),(mp13,),(mp23,),)),
    u1=0.0,u2=0.0,u3=0.0,ur1=0.0,ur2=0.0,ur3=0.0) ## encastre

# Mesh ********************************************************************************

# Seeding
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER,edges=
    skin_instance.edges.findAt((mp12,),(mp13,),(mp23,),),number=skin_seed_number) # seeding by edge
##root_assembly.seedPartInstance(deviationFactor=0.1, minSizeFactor=0.1,
##    regions=(skin_instance,),size=skin_seed_number) # seeding by instance

# Element Shape
root_assembly.setMeshControls(elemShape=skin_element_shape,regions=
    skin_instance.faces.findAt((fc123,),))
# Element Type
mdb.models[model_name].rootAssembly.setElementType(elemTypes=(ElemType(elemCode=skin_element_type1,
    elemLibrary=STANDARD),ElemType(elemCode=skin_element_type2,elemLibrary=STANDARD,
    secondOrderAccuracy=OFF)),regions=(skin_instance.faces.findAt((fc123,)),))
```

```
#Mesh Instance
mdb.models[model_name].rootAssembly.generateMesh(regions=(skin_instance,))

# Field Outputs ********************************************************************************************
if membrane == 1:
    mdb.models[model_name].FieldOutputRequest(name='Field-Output',createStepName=stepname,
                                              variables=('S','U'))
else:
    mdb.models[model_name].FieldOutputRequest(name='Field-Output',createStepName=stepname,
                                              variables=('S','U'),sectionPoints=(1,2,3))

del mdb.models[model_name].fieldOutputRequests['F-Output-1']

# Job ******************************************************************************************************
mdb.Job(atTime=None, contactPrint=OFF, description='',echoPrint=OFF,explicitPrecision=SINGLE,
    getMemoryFromAnalysis=False,historyPrint=OFF,memory=memory_usage,memoryUnits=MEGA_BYTES,
    model=model_name,modelPrint=OFF,multiprocessingMode=DEFAULT,name=job_name,nodalOutputPrecision=FULL,
    numCpus=num_cores,numDomains=num_cores,numGPUs=num_GPUs,queue=None,scratch='',type=ANALYSIS,
    userSubroutine='',waitHours=0, waitMinutes=0)

# Submission ***********************************************************************************************
mdb.jobs[job_name].submit(consistencyChecking=OFF)
mdb.jobs[job_name].waitForCompletion()
```

B.4 Frame Standalone Model

```
# -*- coding: mbcs -*-
import os
from part import*
from material import*
from section import*
from assembly import*
from step import*
from interaction import*
from load import*
from mesh import*
from job import*
from sketch import*
from visualization import*
from connectorBehavior import*
from odbAccess import *
from abaqusConstants import *
from odbMaterial import *
from odbSection import *
session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

# Sets Working Directory ***********************************************************************
os.chdir(path)

# Load variables *******************************************************************************
execfile('Var_frame.py')

# Model name and creation **********************************************************************
mdb.Model(modelType=STANDARD_EXPLICIT, name=model_name)
```

```
# Creates Frame ******************************************************************************
mdb.models[model_name].Part(dimensionality=THREE_D, name='Frame',type=DEFORMABLE_BODY)
frame=mdb.models[model_name].parts['Frame']
frame.DatumPointByCoordinate(p1);  frame.DatumPointByCoordinate(p2)
frame.DatumPointByCoordinate(p3);  frame.DatumPointByCoordinate(p4)
frame.DatumPointByCoordinate(p5);  frame.DatumPointByCoordinate(p6)
frame.DatumPointByCoordinate(p7);  frame.DatumPointByCoordinate(p8)
frame.DatumPointByCoordinate(p9);  frame.DatumPointByCoordinate(p10)
frame.DatumPointByCoordinate(p11); frame.DatumPointByCoordinate(p12)
mdb.models[model_name].ConstrainedSketch(name='__profile__', sheetSize=2.0)
mdb.models[model_name].sketches['__profile__']

frame.WirePolyLine(mergeWire=ON, meshable=ON, points=(
(p1,p2),(p1,p3),(p1,p4),(p1,p5),(p1,p6),(p2,p3),(p3,p4),(p4,p5),(p5,p6),(p6,p2),
        (p12,p11),(p12,p10),(p12,p9),(p12,p8),(p12,p7),(p7,p8),(p8,p9),(p9,p10),(p10,p11),
        (p11,p7),(p2,p11),(p2,p7),(p3,p8),(p3,p7),(p4,p8),(p4,p9),(p5,p9),(p5,p10),(p6,p10),(p6,p11)))

# Materials **********************************************************************************
mdb.models[model_name].Material(name='Frame Material')
mdb.models[model_name].materials['Frame Material'].Density(table=((frame_density,),))
mdb.models[model_name].materials['Frame Material'].Elastic(table=((frame_modulus,frame_poisson),))

# Profiles & Section Assignments *************************************************************
frame_edges=frame.edges.findAt((mp12,),(mp13,),(mp14,),(mp15,),(mp16,),(mp23,),(mp34,),(mp45,),
                               (mp56,),(mp62,),(mp127,),(mp128,),(mp129,),(mp1210,),(mp1211,),
                               (mp78,),(mp89,),(mp910,),(mp1011,),(mp711,),(mp211,),(mp27,),
                               (mp711,),(mp116,),(mp610,),(mp1011,),(mp105,),(mp56,),(mp59,),
                               (mp94,),(mp73,),(mp38,),(mp84,),)

# Frame
if hollow_profile == 1:
    mdb.models[model_name].PipeProfile(name='Frame Beam Profile', r=frame_beam_radius, t=frame_beam_thickness)
else:
    mdb.models[model_name].CircularProfile(name='Frame Beam Profile', r=frame_beam_radius)

mdb.models[model_name].BeamSection(integration=DURING_ANALYSIS,material='Frame Material',name='Frame Beam Section',
        poissonRatio=frame_poisson, profile='Frame Beam Profile',temperatureVar=LINEAR)
frame.SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(
        edges=frame_edges),sectionName='Frame Beam Section',thicknessAssignment=FROM_SECTION)
frame.assignBeamSectionOrientation(method=N1_COSINES,n1=(0.0, 0.0, -1.0),region=Region(edges=frame_edges))

# Assembly ***********************************************************************************
mdb.models[model_name].rootAssembly.Instance(dependent=OFF,name='Frame Instance',part=frame)
frame_instance = mdb.models[model_name].rootAssembly.instances['Frame Instance']
root_assembly = mdb.models[model_name].rootAssembly

frame_edges_instance=frame_instance.edges.findAt((mp12,),(mp13,),(mp14,),(mp15,),(mp16,),(mp23,),(mp34,),(mp45,),
                                      (mp56,),(mp62,),(mp127,),(mp128,),(mp129,),(mp1210,),(mp1211,),
                                      (mp78,),(mp89,),(mp910,),(mp1011,),(mp711,),(mp211,),(mp27,),
                                      (mp711,),(mp116,),(mp610,),(mp1011,),(mp105,),(mp56,),(mp59,),
                                      (mp94,),(mp73,),(mp38,),(mp84,),)

# Spherical System ***************************************************************************
spherical_id = root_assembly.DatumCsysByThreePoints(coordSysType=SPHERICAL,name='Spherical',
               origin=(0.0,0.0,0.0),point1=(1.0,0.0,0.0),point2=(0.0,1.0,0.0)).id

#Step ****************************************************************************************
if buckle == 1:
```

```
    mdb.models[model_name].BuckleStep(maxIterations=buck_max_Iter,name=stepname,
    numEigen=buck_num_Eigen,previous='Initial',vectors=buck_num_vectors)
elif stabilization == 1:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,previous='Initial',
    continueDampingFactors=False,adaptiveDampingRatio=stabilization_ratio,matrixStorage=UNSYMMETRIC,
    stabilizationMagnitude=stabilization_magn,stabilizationMethod=DISSIPATED_ENERGY_FRACTION,
    timeIncrementationMethod=increment_method)
elif step_type == 1:
    mdb.models[model_name].StaticRiksStep(initialArcInc=initial_ArcInc,maxLPF=max_LPF,
    minArcInc=min_ArcInc,maxArcInc=max_ArcInc,maxNumInc=maxnuminc,name=stepname,matrixStorage=UNSYMMETRIC,
    nlgeom=nonlinear_effects,timeIncrementationMethod=increment_method,previous='Initial')
else:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,
    timeIncrementationMethod=increment_method,previous='Initial')

if buckle == 1:
    buckle = 1
else:
    mdb.models[model_name].steps[stepname].control.setValues(
    allowPropagation=OFF, discontinuous=ON, displacementField=(0.005, 0.01,
    0.0, 0.0, 0.02, 1e-05, 0.001, 1e-08, 1.0, 1e-05, 1e-12),
    hydrostaticFluidPressureField=(0.005, 0.01, 0.0, 0.0, 0.02, 1e-05, 0.001,
    1e-08, 1.0, 1e-05), lineSearch=(10.0, 1.0, 0.0001, 0.25, 0.01),
    resetDefaultValues=OFF, rotationField=(0.005, 0.01, 0.0, 0.0, 0.02, 1e-05,
    0.001, 1e-08, 1.0, 1e-05), timeIncrementation=(8.0, 10.0, 9.0, 16.0, 10.0,
    4.0, 12.0, 20.0, 6.0, 3.0, 50.0, 0.5, 0.5, 0.75, 0.85, 0.25, 0.25, 1.5,
    0.75))

#BC ******************************************************************************
mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=
    stepname,distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BCbottom',
    region=Region(vertices=frame_instance.vertices.findAt((p12,),)),
    u1=0.0,u2=0.0,u3=0.0,ur1=0.0,ur2=0.0,ur3=0.0) ## Bottom BC (all DOFs)
mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=
    stepname,distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BCtop',
    region=Region(vertices=frame_instance.vertices.findAt((p1,),)),u1=0.0,u2=0.0) ## Bottom BC (u1=u2=0)

# Mesh ********************************************************************************

# Beam
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER,
    edges=frame_edges_instance,number=frame_seed_number)
mdb.models[model_name].rootAssembly.setElementType(elemTypes=(ElemType(
    elemCode=frame_element_type,elemLibrary=STANDARD),),regions=(frame_edges_instance,))

#Meshes Instances
mdb.models[model_name].rootAssembly.generateMesh(regions=(frame_instance,))

# Coupling Constraints and Forces ***************************************************************************
mp = [mp12,mp13,mp23, mp13,mp14,mp34, mp14,mp15,mp45, mp15,mp16,mp56, mp16,mp12,mp62, mp127,mp128,mp78,
      mp128,mp129,mp89, mp129,mp1210,mp910, mp1210,mp1211,mp1110,mp127,mp1211,mp711,mp211,mp27,mp711,
      mp112,mp116,mp26, mp610,mp611,mp1011, mp105,mp106,mp56, mp59,mp510,mp910, mp94,mp95,mp45,
      mp48,mp49,mp89, mp83,mp84,mp34, mp37,mp38,mp78, mp72,mp73,mp23]
fc1 = [fc123,fc134,fc145,fc156,fc162,fc1278,fc1289,fc12910,fc121011,fc12117,fc2711,fc1126,fc61011,
       fc1056,fc5910,fc945,fc489,fc834,fc378,fc723]
r = p1[2]
```

```
el = r*sqrt(50-10*sqrt(5))/5
A = el**2*sqrt(3)/4
F = P*A
g = 0;
for i in range(0,20):
    # Creates Reference Points
    root_assembly.ReferencePoint(point=fc1[i])
    # Creates Sets for Coupling using Reference Points
    set1 = root_assembly.Set(name='CouplingSet'+str(i), referencePoints=(
           root_assembly.referencePoints.findAt(fc1[i]),))
    # Create Surfaces for Coupling using Frame and Skin Edges
    ed   = frame_instance.edges.findAt((mp[g],),(mp[g+1],),(mp[g+2],),)
    set2 = mdb.models[model_name].rootAssembly.Surface(circumEdges=ed, name='FrameSet'+str(i))
    # Coupling Constraints
    mdb.models[model_name].Coupling(controlPoint=set1,couplingType=DISTRIBUTING,
    influenceRadius=WHOLE_SURFACE, localCsys=None, name='CoupConstraint'+str(i),
    surface=set2, u1=ON, u2=ON, u3=ON, ur1=OFF, ur2=OFF, ur3=OFF, weightingMethod=UNIFORM)
    g = g + 3
    # Load or BC
    if disp_control == 1:
        mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=
            stepname,distributionType=UNIFORM,fieldName='',fixed=OFF,
            localCsys=root_assembly.datums[spherical_id],name='BC Disp Controlled'+str(i),
            region=set1,u1=d)
    else:
        mdb.models[model_name].ConcentratedForce(cf1=-F,createStepName=stepname,
            distributionType=UNIFORM,field='', localCsys=root_assembly.datums[spherical_id],name='Load'+str(i),region=set1)

# Field Outputs ******************************************************************************************
if buckle == 1:
    mdb.models[model_name].FieldOutputRequest(name='FO-WholeModel',createStepName=stepname,
    variables=('U','S'))
else:
    mdb.models[model_name].FieldOutputRequest(name='FO-WholeModel',createStepName=stepname,
    variables=('U','S','NFORC','E','P','RF'))

del mdb.models[model_name].fieldOutputRequests['F-Output-1']

# Job ****************************************************************************************************
mdb.Job(atTime=None, contactPrint=OFF, description='',echoPrint=OFF,explicitPrecision=SINGLE,
    getMemoryFromAnalysis=False,historyPrint=OFF,memory=memory_usage,memoryUnits=MEGA_BYTES,
    model=model_name,modelPrint=OFF,multiprocessingMode=DEFAULT,name=job_name,nodalOutputPrecision=FULL,
    numCpus=num_cores,numDomains=num_cores,numGPUs=num_GPUs,queue=None,scratch='',type=ANALYSIS,
    userSubroutine='',waitHours=0, waitMinutes=0)

# Submission *********************************************************************************************
mdb.jobs[job_name].submit(consistencyChecking=OFF)
mdb.jobs[job_name].waitForCompletion()
```

B.5 Icosahedron

```
# -*- coding: mbcs -*-
import os
from part import*
from material import*
```

```
from section import*
from assembly import*
from step import*
from interaction import*
from load import*
from mesh import*
from job import*
from sketch import*
from visualization import*
from connectorBehavior import*
from odbAccess import *
from abaqusConstants import *
from odbMaterial import *
from odbSection import *
session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

# Sets Working Directory ********************************************************************
os.chdir(path)

# Load variables ****************************************************************************
execfile('Var_icosahedron.py')

# Model name and creation *******************************************************************
mdb.Model(modelType=STANDARD_EXPLICIT, name=model_name)

# Creates Frame *****************************************************************************
mdb.models[model_name].Part(dimensionality=THREE_D, name='Frame',type=DEFORMABLE_BODY)
frame=mdb.models[model_name].parts['Frame']
frame.DatumPointByCoordinate(p1);  frame.DatumPointByCoordinate(p2)
frame.DatumPointByCoordinate(p3);  frame.DatumPointByCoordinate(p4)
frame.DatumPointByCoordinate(p5);  frame.DatumPointByCoordinate(p6)
frame.DatumPointByCoordinate(p7);  frame.DatumPointByCoordinate(p8)
frame.DatumPointByCoordinate(p9);  frame.DatumPointByCoordinate(p10)
frame.DatumPointByCoordinate(p11); frame.DatumPointByCoordinate(p12)
mdb.models[model_name].ConstrainedSketch(name='__profile__', sheetSize=2.0)
mdb.models[model_name].sketches['__profile__']

frame.WirePolyLine(mergeWire=ON, meshable=ON, points=(
(p1,p2),(p1,p3),(p1,p4),(p1,p5),(p1,p6),(p2,p3),(p3,p4),(p4,p5),(p5,p6),(p6,p2),
        (p12,p11),(p12,p10),(p12,p9),(p12,p8),(p12,p7),(p7,p8),(p8,p9),(p9,p10),(p10,p11),
        (p11,p7),(p2,p11),(p2,p7),(p3,p8),(p3,p7),(p4,p8),(p4,p9),(p5,p9),(p5,p10),(p6,p10),(p6,p11)))

# Creates Skin ******************************************************************************
mdb.models[model_name].Part(dimensionality=THREE_D, name='Skin', type=DEFORMABLE_BODY)
skin=mdb.models[model_name].parts['Skin']
mdb.models[model_name].ConstrainedSketch(name='__profile__', sheetSize=2.0)
mdb.models[model_name].sketches['__profile__']

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p1,p2),(p1,p3),(
p2,p3)))
e12=skin.edges.findAt(mp12); e13=skin.edges.findAt(mp13); e23=skin.edges.findAt(mp23)
skin.CoverEdges((e12,e13,e23), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p1,p3),(p1,p4),(
p3,p4)))
e13=skin.edges.findAt(mp13); e14=skin.edges.findAt(mp14); e34=skin.edges.findAt(mp34)
skin.CoverEdges((e13,e14,e34), tryAnalytical=True)
```

```
skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p1,p4),(p1,p5),(
p4,p5)))
e14=skin.edges.findAt(mp14); e15=skin.edges.findAt(mp15); e45=skin.edges.findAt(mp45)
skin.CoverEdges((e14,e15,e45), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p1,p5),(p1,p6),(
p5,p6)))
e15=skin.edges.findAt(mp15); e16=skin.edges.findAt(mp16); e56=skin.edges.findAt(mp56)
skin.CoverEdges((e15,e16,e56), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p1,p6),(p1,p2),(
p6,p2)))
e16=skin.edges.findAt(mp16); e12=skin.edges.findAt(mp12); e62=skin.edges.findAt(mp62)
skin.CoverEdges((e16,e12,e62), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p12,p7),(p12,p8),(
p7,p8)))
e127=skin.edges.findAt(mp127); e128=skin.edges.findAt(mp128); e78=skin.edges.findAt(mp78)
skin.CoverEdges((e127,e128,e78), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p12,p8),(p12,p9),(
p8,p9)))
e128=skin.edges.findAt(mp128); e129=skin.edges.findAt(mp129); e89=skin.edges.findAt(mp89)
skin.CoverEdges((e128,e129,e89), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p12,p9),(p12,p10),(
p9,p10)))
e129=skin.edges.findAt(mp129); e1210=skin.edges.findAt(mp1210); e910=skin.edges.findAt(mp910)
skin.CoverEdges((e129,e1210,e910), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p12,p10),(p12,p11),(
p11,p10)))
e1210=skin.edges.findAt(mp1210); e1211=skin.edges.findAt(mp1211); e1110=skin.edges.findAt(mp1110)
skin.CoverEdges((e1210,e1211,e1110), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p12,p7),(p12,p11),(
p7,p11)))
e127=skin.edges.findAt(mp127); e1211=skin.edges.findAt(mp1211); e711=skin.edges.findAt(mp711)
skin.CoverEdges((e127,e1211,e711), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p2,p11),(p2,p7),(
p7,p11)))
e211=skin.edges.findAt(mp211); e27=skin.edges.findAt(mp27); e711=skin.edges.findAt(mp711)
skin.CoverEdges((e211,e27,e711), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p11,p2),(p11,p6),(
p2,p6)))
e112=skin.edges.findAt(mp112); e116=skin.edges.findAt(mp116); e26=skin.edges.findAt(mp26)
skin.CoverEdges((e112,e116,e26), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p6,p10),(p6,p11),(
p10,p11)))
e610=skin.edges.findAt(mp610); e611=skin.edges.findAt(mp611); e1011=skin.edges.findAt(mp1011)
skin.CoverEdges((e610,e611,e1011), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p10,p5),(p10,p6),(
p5,p6)))
e105=skin.edges.findAt(mp105); e106=skin.edges.findAt(mp106); e56=skin.edges.findAt(mp56)
```

```
skin.CoverEdges((e105,e106,e56), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p5,p9),(p5,p10),(
p9,p10)))
e59=skin.edges.findAt(mp59); e510=skin.edges.findAt(mp510); e910=skin.edges.findAt(mp910)
skin.CoverEdges((e59,e510,e910), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p9,p4),(p9,p5),(
p4,p5)))
e94=skin.edges.findAt(mp94); e95=skin.edges.findAt(mp95); e45=skin.edges.findAt(mp45)
skin.CoverEdges((e94,e95,e45), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p4,p8),(p4,p9),(
p8,p9)))
e48=skin.edges.findAt(mp48); e49=skin.edges.findAt(mp49); e89=skin.edges.findAt(mp89)
skin.CoverEdges((e48,e49,e89), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p8,p3),(p8,p4),(
p3,p4)))
e83=skin.edges.findAt(mp83); e84=skin.edges.findAt(mp84); e34=skin.edges.findAt(mp34)
skin.CoverEdges((e83,e84,e34), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p3,p7),(p3,p8),(
p7,p8)))
e37=skin.edges.findAt(mp37); e38=skin.edges.findAt(mp38); e78=skin.edges.findAt(mp78)
skin.CoverEdges((e37,e38,e78), tryAnalytical=True)

skin.WirePolyLine(mergeWire=ON,meshable=ON, points=((p7,p2),(p7,p3),(
p2,p3)))
e72=skin.edges.findAt(mp72); e73=skin.edges.findAt(mp73); e23=skin.edges.findAt(mp23)
skin.CoverEdges((e72,e73,e23), tryAnalytical=True)

# Creates stiffners ******************************************************************
if stiff_select == 1:
    for i in range(0,len(k)):
        frame.DatumPointByCoordinate(fc[i])
        frame.WirePolyLine(mergeWire=ON,meshable=ON,
            points=((p_array[k[i][0]-1],fc[i]),(p_array[k[i][1]-1],fc[i]),(p_array[k[i][2]-1],
                        fc[i])))

# Creates Center Rays ****************************************************************
if rays_select == 1:
    mdb.models[model_name].Part(dimensionality=THREE_D, name='Rays', type=DEFORMABLE_BODY)
    rays=mdb.models[model_name].parts['Rays']
    fc = [(fc123),(fc134),(fc145),(fc156),(fc162),(fc1278),(fc1289),
          (fc12910),(fc121011),(fc12117),(fc723),(fc378),(fc834),
          (fc489),(fc945),(fc5910),(fc1056),(fc61011),(fc1126),(fc2711)]
    c = (0,0,0)
    rays.DatumPointByCoordinate(c)
    for i in range(0,len(fc)):
        rays.DatumPointByCoordinate(fc[i]);
        rays.WirePolyLine(mergeWire=ON,meshable=ON,points=((c,fc[i])))

# Materials **************************************************************************
mdb.models[model_name].Material(name='Frame Material')
mdb.models[model_name].materials['Frame Material'].Density(table=((frame_density,),))
mdb.models[model_name].materials['Frame Material'].Elastic(table=((frame_modulus,frame_poisson),))
mdb.models[model_name].Material(name='Skin Material')
```

```
mdb.models[model_name].materials['Skin Material'].Density(table=((skin_density,),))
mdb.models[model_name].materials['Skin Material'].Elastic(table=((skin_modulus,skin_poisson),))
mdb.models[model_name].Material(name='Rays Material')
mdb.models[model_name].materials['Rays Material'].Density(table=((rays_density,),))
mdb.models[model_name].materials['Rays Material'].Elastic(table=((rays_modulus,rays_poisson),))
mdb.models[model_name].Material(name='Stiffners Material')
mdb.models[model_name].materials['Stiffners Material'].Density(table=((stiff_density,),))
mdb.models[model_name].materials['Stiffners Material'].Elastic(table=((stiff_modulus,stiff_poisson),))

# Profiles & Section Assignments ********************************************************
skin_faces=skin.faces.findAt((fc123,),(fc134,),(fc145,),(fc156,),(fc162,),(fc1278,),(fc1289,),
                             (fc12910,),(fc121011,),(fc12117,),(fc723,),(fc378,),(fc834,),
                             (fc489,),(fc945,),(fc5910,),(fc1056,),(fc61011,),(fc1126,),(fc2711,),)
frame_edges=frame.edges.findAt((mp12,),(mp13,),(mp14,),(mp15,),(mp16,),(mp23,),(mp34,),(mp45,),
                               (mp56,),(mp62,),(mp127,),(mp128,),(mp129,),(mp1210,),(mp1211,),
                               (mp78,),(mp89,),(mp910,),(mp1011,),(mp711,),(mp211,),(mp27,),
                               (mp711,),(mp116,),(mp610,),(mp1011,),(mp105,),(mp56,),(mp59,),
                               (mp94,),(mp73,),(mp38,),(mp84,),)

# Frame
if hollow_profile == 1:
    mdb.models[model_name].PipeProfile(name='Frame Beam Profile', r=frame_beam_radius,
                                                                  t=frame_beam_thickness)
else:
    mdb.models[model_name].CircularProfile(name='Frame Beam Profile', r=frame_beam_radius)

mdb.models[model_name].BeamSection(integration=DURING_ANALYSIS,material='Frame Material',
        name='Frame Beam Section',
        poissonRatio=frame_poisson, profile='Frame Beam Profile',temperatureVar=LINEAR)
frame.SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(
        edges=frame_edges),sectionName='Frame Beam Section',thicknessAssignment=FROM_SECTION)
frame.assignBeamSectionOrientation(method=N1_COSINES,n1=(0.0, 0.0, -1.0),region=
          Region(edges=frame_edges))

# Skin
if no_stiffness_skin == 1:
    mdb.models[model_name].SurfaceSection(density=skin_density,name='Skin Section',useDensity=ON)
elif membrane == 1:
    mdb.models[model_name].MembraneSection(material=
        'Skin Material', name='Skin Section', poisson=skin_poisson,poissonDefinition=VALUE,
        thickness=skin_thickness,thicknessField='',thicknessType=UNIFORM)
else:
    mdb.models[model_name].HomogeneousShellSection(idealization=skin_section_idealization,
        material='Skin Material',name='Skin Section',poisson=skin_poisson,poissonDefinition=VALUE,
preIntegrate=ON,thickness=skin_thickness,thicknessField='',thicknessModulus=None,
        thicknessType=UNIFORM, useDensity=ON)

skin.SectionAssignment(offset=0.0,offsetField='',offsetType=skin_section_location,region=Region(
        faces=skin_faces),sectionName='Skin Section',thicknessAssignment=FROM_SECTION)

# Rays
if rays_select == 1:
    rays_edges=rays.edges.findAt((fc123,),(fc134,),(fc145,),(fc156,),(fc162,),(fc1278,),(fc1289,),
                                 (fc12910,),(fc121011,),(fc12117,),(fc723,),(fc378,),(fc834,),
                                 (fc489,),(fc945,),(fc5910,),(fc1056,),(fc61011,),(fc1126,),(fc2711,),)
    if hollow_profile_rays == 1:
        mdb.models[model_name].PipeProfile(name='Rays Beam Profile', r=rays_beam_radius,
                         t=rays_beam_thickness)
```

```
    else:
        mdb.models[model_name].CircularProfile(name='Rays Beam Profile', r=rays_beam_radius)

    mdb.models[model_name].BeamSection(integration=DURING_ANALYSIS,material='Rays Material',
                                       name='Rays Beam Section',
        poissonRatio=rays_poisson, profile='Rays Beam Profile',temperatureVar=LINEAR)
    rays.SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(
        edges=rays_edges),sectionName='Rays Beam Section',thicknessAssignment=FROM_SECTION)
    rays.assignBeamSectionOrientation(method=N1_COSINES,n1=(0.0, 0.0, -1.0),region=
                                Region(edges=rays_edges))

# Stiffners
if stiff_select == 1:
    stiff_edges=frame.edges.findAt((np[0],),(np[1],),(np[2],),(np[3],),(np[4],),(np[5],),
            (np[6],),(np[7],),(np[8],),(np[9],),(np[10],),
            (np[11],),(np[12],),(np[13],),(np[14],),(np[15],),(np[16],),(np[17],),(np[18],),
            (np[19],),(np[20],),
            (np[21],),(np[22],),(np[23],),(np[24],),(np[25],),(np[26],),(np[27],),(np[28],),
            (np[29],),(np[30],),
            (np[31],),(np[32],),(np[33],),(np[34],),(np[35],),(np[36],),(np[37],),(np[38],),
            (np[39],),(np[40],),
            (np[41],),(np[42],),(np[43],),(np[44],),(np[45],),(np[46],),(np[47],),(np[48],),
            (np[49],),(np[50],),
            (np[51],),(np[52],),(np[53],),(np[54],),(np[55],),(np[56],),(np[57],),(np[58],),
            (np[59],),)
    if hollow_profile_stiff == 1:
        mdb.models[model_name].PipeProfile(name='Stiffners Beam Profile', r=stiff_beam_radius,
                    t=stiff_beam_thickness)
    else:
        mdb.models[model_name].CircularProfile(name='Stiffners Beam Profile', r=stiff_beam_radius)

    mdb.models[model_name].BeamSection(integration=DURING_ANALYSIS,material='Stiffners Material',
                                     name='Stiffners Beam Section',
        poissonRatio=rays_poisson, profile='Stiffners Beam Profile',temperatureVar=LINEAR)
    frame.SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(
        edges=stiff_edges),sectionName='Stiffners Beam Section',thicknessAssignment=FROM_SECTION)
    frame.assignBeamSectionOrientation(method=N1_COSINES,n1=(0.0, 0.0, -1.0),region=
                            Region(edges=stiff_edges))

# Assembly ********************************************************************************
mdb.models[model_name].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models[model_name].rootAssembly.Instance(dependent=OFF,name='Frame Instance',part=frame)
mdb.models[model_name].rootAssembly.Instance(dependent=OFF,name='Skin Instance',part=skin)
frame_instance = mdb.models[model_name].rootAssembly.instances['Frame Instance']
skin_instance = mdb.models[model_name].rootAssembly.instances['Skin Instance']
root_assembly = mdb.models[model_name].rootAssembly

if rays_select == 1:
    mdb.models[model_name].rootAssembly.Instance(dependent=OFF,name='Rays Instance',part=rays)
    rays_instance = mdb.models[model_name].rootAssembly.instances['Rays Instance']

skin_faces_instance=skin_instance.faces.findAt((fc123,),(fc134,),(fc145,),(fc156,),
                (fc162,),(fc1278,),(fc1289,),
               (fc12910,),(fc121011,),(fc12117,),(fc723,),(fc378,),(fc834,),
                (fc489,),(fc945,),(fc5910,),(fc1056,),(fc61011,),(fc1126,),(fc2711,))
skin_edges_instance= skin_instance.edges.findAt((mp12,),(mp13,),(mp14,),
                 (mp15,),(mp16,),(mp23,),(mp34,),(mp45,),
                  (mp56,),(mp62,),(mp127,),(mp128,),(mp129,),(mp1210,),(mp1211,),
```

```
                    (mp78,),(mp89,),(mp910,),(mp1011,),(mp711,),(mp211,),(mp27,),
                     (mp711,),(mp116,),(mp610,),(mp1011,),(mp105,),(mp56,),(mp59,),
                     (mp94,),(mp73,),(mp38,),(mp84,),)
frame_edges_instance=frame_instance.edges.findAt((mp12,),(mp13,),(mp14,),
                      (mp15,),(mp16,),(mp23,),(mp34,),(mp45,),
                      (mp56,),(mp62,),(mp127,),(mp128,),(mp129,),(mp1210,),(mp1211,),
                      (mp78,),(mp89,),(mp910,),(mp1011,),(mp711,),(mp211,),(mp27,),
                       (mp711,),(mp116,),(mp610,),(mp1011,),(mp105,),(mp56,),(mp59,),
                       (mp94,),(mp73,),(mp38,),(mp84,),)
if rays_select == 1:
    rays_edges_instance=rays_instance.edges.findAt((fc123,),(fc134,),(fc145,),
            (fc156,),(fc162,),(fc1278,),
(fc1289,),(fc12910,),(fc121011,),(fc12117,),(fc723,),(fc378,),(fc834,),
(fc489,),(fc945,),(fc5910,),(fc1056,),(fc61011,),(fc1126,),(fc2711,),)
    rays_vertices_instance=rays_instance.vertices.findAt((fc123,),(fc134,),
(fc145,),(fc156,),(fc162,),(fc1278,),(fc1289,),
(fc12910,),(fc121011,),(fc12117,),(fc723,),(fc378,),(fc834,),
(fc489,),(fc945,),(fc5910,),(fc1056,),(fc61011,),(fc1126,),(fc2711,),)

if stiff_select == 1:
    stiff_edges_instance=frame.edges.findAt((np[0],),(np[1],),(np[2],),(np[3],),(np[4],),(np[5],),
            (np[6],),(np[7],),(np[8],),(np[9],),(np[10],),
(np[11],),(np[12],),(np[13],),(np[14],),(np[15],),(np[16],),(np[17],),(np[18],),(np[19],),(np[20],),
(np[21],),(np[22],),(np[23],),(np[24],),(np[25],),(np[26],),(np[27],),(np[28],),(np[29],),(np[30],),
 (np[31],),(np[32],),(np[33],),(np[34],),(np[35],),(np[36],),(np[37],),(np[38],),(np[39],),(np[40],),
(np[41],),(np[42],),(np[43],),(np[44],),(np[45],),(np[46],),(np[47],),(np[48],),(np[49],),(np[50],),
(np[51],),(np[52],),(np[53],),(np[54],),(np[55],),(np[56],),(np[57],),(np[58],),(np[59],),)
    stiff_frame_edges_instance=frame.edges.findAt((np[0],),(np[1],),(np[2],),(np[3],),(np[4],),(np[5],),
(np[6],),(np[7],),(np[8],),(np[9],),(np[10],),
(np[11],),(np[12],),(np[13],),(np[14],),(np[15],),(np[16],),(np[17],),(np[18],),(np[19],),(np[20],),
    (np[21],),(np[22],),(np[23],),(np[24],),(np[25],),(np[26],),(np[27],),(np[28],),(np[29],),(np[30],),
(np[31],),(np[32],),(np[33],),(np[34],),(np[35],),(np[36],),(np[37],),(np[38],),(np[39],),(np[40],),
   (np[41],),(np[42],),(np[43],),(np[44],),(np[45],),(np[46],),(np[47],),(np[48],),(np[49],),(np[50],),
(np[51],),(np[52],),(np[53],),(np[54],),(np[55],),(np[56],),(np[57],),(np[58],),(np[59],),
   (mp12,),(mp13,),(mp14,),(mp15,),(mp16,),(mp23,),(mp34,),(mp45,),
 (mp56,),(mp62,),(mp127,),(mp128,),(mp129,),(mp1210,),(mp1211,),
       (mp78,),(mp89,),(mp910,),(mp1011,),(mp711,),(mp211,),(mp27,),
 (mp711,),(mp116,),(mp610,),(mp1011,),(mp105,),(mp56,),(mp59,),
(mp94,),(mp73,),(mp38,),(mp84,),)
    stiff_frame_vertices_instance=frame.edges.findAt((fc123,),(fc134,),(fc145,),(fc156,),(fc162,),
(fc1278,),(fc1289,),(fc12910,),(fc121011,),(fc12117,),(fc723,),(fc378,),(fc834,),
            (fc489,),(fc945,),(fc5910,),(fc1056,),(fc61011,),(fc1126,),(fc2711,),
            (p1,),(p2,),(p3,),(p4,),(p5,),(p6,),(p7,),(p8,),(p9,),(p10,),(p11,),(p12,),)

# Spherical System ***************************************************************************************
spherical_id = root_assembly.DatumCsysByThreePoints(coordSysType=SPHERICAL,name='Spherical',
               origin=(0.0,0.0,0.0),point1=(1.0,0.0,0.0),point2=(0.0,1.0,0.0)).id

# Spherical Datum Convertion *****************************************************************
    ##datumid_skin=skin.DatumCsysByThreePoints(coordSysType=SPHERICAL,name='Spherical',
    ##        origin=(0.0,0.0,0.0),point1=(1.0,0.0,0.0),point2=(0.0,1.0,0.0)).id
    ##skin.MaterialOrientation(
    ##    additionalRotationField='',additionalRotationType=ROTATION_NONE,angle=0.0,axis=AXIS_3,
    ##     fieldName='',
    ##    localCsys=skin.datums[datumid_skin],orientationType=SYSTEM,region=Region(faces=skin_faces))
    ##
    ##datumid_frame=frame.DatumCsysByThreePoints(coordSysType=SPHERICAL,name='Spherical',
    ##        origin=(0.0,0.0,0.0),point1=(1.0,0.0,0.0),point2=(0.0,1.0,0.0)).id
```

```
    ##frame.MaterialOrientation(
    ##    additionalRotationField='',additionalRotationType=ROTATION_NONE,angle=0.0,axis=AXIS_3,
    ##    fieldName='',
    ##    localCsys=frame.datums[datumid_frame],orientationType=SYSTEM,region=Region(edges=frame_edges))
    ##
    ##mdb.models[model_name].rootAssembly.regenerate()

# Constraint ******************************************************************************
if stiff_select == 1:
    # Add constraint to connect stiffeners to the skin
else:
    mdb.models[model_name].Tie(adjust=ON, master=Region(edges=frame_edges_instance),
        name='Tie Constraint 1', positionToleranceMethod=COMPUTED,constraintEnforcement=NODE_TO_SURFACE,
        slave=Region(edges=skin_edges_instance),thickness=ON, tieRotations=rotations)

if rays_select == 1:
    mdb.models[model_name].Tie(adjust=ON, master=Region(faces=skin_faces_instance),
    name='Tie Constraint 2', positionToleranceMethod=COMPUTED,constraintEnforcement=NODE_TO_SURFACE,
    slave=Region(vertices=rays_vertices_instance),thickness=ON, tieRotations=rotations)

#Step *************************************************************************************
if buckle == 1:
    mdb.models[model_name].BuckleStep(maxIterations=buck_max_Iter,name=stepname,
    numEigen=buck_num_Eigen,previous='Initial',vectors=buck_num_vectors)
elif stabilization == 1:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,previous='Initial',
    continueDampingFactors=False,adaptiveDampingRatio=stabilization_ratio,matrixStorage=UNSYMMETRIC,
    stabilizationMagnitude=stabilization_magn,stabilizationMethod=DISSIPATED_ENERGY_FRACTION,
    timeIncrementationMethod=increment_method)
elif step_type == 1:
    mdb.models[model_name].StaticRiksStep(initialArcInc=initial_ArcInc,maxLPF=max_LPF,
    minArcInc=min_ArcInc,maxArcInc=max_ArcInc,maxNumInc=maxnuminc,name=stepname,matrixStorage=UNSYMMETRIC,
    nlgeom=nonlinear_effects,timeIncrementationMethod=increment_method,previous='Initial')
else:
    mdb.models[model_name].StaticStep(initialInc=initial_inc,maxInc=max_inc,
    minInc=min_inc,maxNumInc=maxnuminc,name=stepname,nlgeom=nonlinear_effects,
    timeIncrementationMethod=increment_method,previous='Initial')

if buckle == 1:
    buckle = 1
else:
    mdb.models[model_name].steps[stepname].control.setValues(
    allowPropagation=OFF, discontinuous=ON, displacementField=(0.005, 0.01,
    0.0, 0.0, 0.02, 1e-05, 0.001, 1e-08, 1.0, 1e-05, 1e-12),
    hydrostaticFluidPressureField=(0.005, 0.01, 0.0, 0.0, 0.02, 1e-05, 0.001,
    1e-08, 1.0, 1e-05), lineSearch=(10.0, 1.0, 0.0001, 0.25, 0.01),
    resetDefaultValues=OFF, rotationField=(0.005, 0.01, 0.0, 0.0, 0.02, 1e-05,
    0.001, 1e-08, 1.0, 1e-05), timeIncrementation=(8.0, 10.0, 9.0, 16.0, 10.0,
    4.0, 12.0, 20.0, 6.0, 3.0, 50.0, 0.5, 0.5, 0.75, 0.85, 0.25, 0.25, 1.5,
    0.75))

#BC ***************************************************************************************
mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=
    stepname,distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BCbottom',
    region=Region(vertices=frame_instance.vertices.findAt((p12,),)),
    u1=0.0,u2=0.0) #,u3=0.0,ur1=0.0,ur2=0.0,ur3=0.0) ## Bottom BC (all DOFs)
mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=
```

```
    stepname,distributionType=UNIFORM,fieldName='',fixed=OFF,localCsys=None,name='BCtop',
    region=Region(vertices=frame_instance.vertices.findAt((p1,),)),u1=0.0,u2=0.0) ## Bottom BC (u1=u2=0)

# Mesh ********************************************************************************

# Beam
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER,
    edges=frame_edges_instance,number=frame_seed_number)
mdb.models[model_name].rootAssembly.setElementType(elemTypes=(ElemType(
    elemCode=frame_element_type,elemLibrary=STANDARD),),regions=(frame_edges_instance,))

# Skin
mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER,
    edges=skin_edges_instance,number=skin_seed_number)
mdb.models[model_name].rootAssembly.setMeshControls(elemShape=skin_element_shape,
            regions=skin_faces_instance)
mdb.models[model_name].rootAssembly.setElementType(elemTypes=(ElemType(elemCode=skin_element_type1,
    elemLibrary=STANDARD),ElemType(elemCode=skin_element_type2,elemLibrary=STANDARD,
            secondOrderAccuracy=OFF)),    regions=(skin_faces_instance,))

# Rays
if rays_select == 1:
    mdb.models[model_name].rootAssembly.seedEdgeByNumber(constraint=FINER,
    edges=rays_edges_instance,number=rays_seed_number)
    mdb.models[model_name].rootAssembly.setElementType(elemTypes=(ElemType(
    elemCode=rays_element_type,elemLibrary=STANDARD),),regions=(rays_edges_instance,))

# Stiffners
if stiff_select == 1:
    for i in range(0,len(np)):
        root_assembly.seedEdgeByNumber(constraint=FINER,
            edges=frame_instance.edges.findAt((np[i],)),number=stiff_seed_number)
        root_assembly.setElementType(elemTypes=(ElemType(elemCode=stiff_element_type, elemLibrary=
STANDARD),),regions=(frame_instance.edges.findAt((np[i],)),))

#Meshes Instances
mdb.models[model_name].rootAssembly.generateMesh(regions=(frame_instance,))
mdb.models[model_name].rootAssembly.generateMesh(regions=(skin_instance,))
if rays_select == 1:
    mdb.models[model_name].rootAssembly.generateMesh(regions=(rays_instance,))

# Load / Displacement Controlled BC ***************************************************************
fc = [(fc123),(fc134),(fc145),(fc156),(fc162),(fc1278),(fc1289),
      (fc12910),(fc121011),(fc12117),(fc723),(fc378),(fc834),
      (fc489),(fc945),(fc5910),(fc1056),(fc61011),(fc1126),(fc2711)]
allNodes = skin_instance.nodes
delta = 1e-6

if disp_control == 1:
    for i in range(0,len(fc)):
        root_assembly.ReferencePoint(point=fc[i])
        myNodes = allNodes.getByBoundingSphere(center = fc[i],radius=delta)
        root_assembly.Set(name='FC-Set'+str(i), nodes=myNodes)
        mdb.models[model_name].DisplacementBC(amplitude=UNSET,createStepName=
            stepname,distributionType=UNIFORM,fieldName='',fixed=OFF,
            localCsys=root_assembly.datums[spherical_id],name='BC Disp Controlled'+str(i),
            region=root_assembly.sets['FC-Set'+str(i)],
            u1=d)
```

```
else:
    #Pressure Load
    mdb.models[model_name].Pressure(amplitude=UNSET,createStepName=
    stepname,distributionType=UNIFORM,field='',magnitude=P,name='Pressure Towards Center 1',
    region=Region(side1Faces=skin_instance.faces.findAt((fc1278,),(fc1289,),(fc12910,),(fc121011,),
                                                        (fc12117,),(fc834,),(fc945,),(fc1056,),)))
    mdb.models[model_name].Pressure(amplitude=UNSET,createStepName=
        stepname,distributionType=UNIFORM,field='',magnitude=P,name='Pressure Towards Center 2',
        region=Region(side2Faces=skin_instance.faces.findAt((fc123,),(fc134,),(fc145,),(fc156,),(fc162,),
(fc723,),(fc378,),(fc489,),(fc5910,),(fc61011,),(fc1126,),(fc2711,),)))

# Field Outputs ********************************************************************************************
mdb.models[model_name].FieldOutputRequest(name='FO-WholeModel',createStepName=stepname,
    variables=('U','S','NFORC','E','P'))
del mdb.models[model_name].fieldOutputRequests['F-Output-1']

# Job ******************************************************************************************************
mdb.Job(atTime=None, contactPrint=OFF, description='',echoPrint=OFF,explicitPrecision=SINGLE,
    getMemoryFromAnalysis=False,historyPrint=OFF,memory=memory_usage,memoryUnits=MEGA_BYTES,
    model=model_name,modelPrint=OFF,multiprocessingMode=DEFAULT,name=job_name,nodalOutputPrecision=FULL,
    numCpus=num_cores,numDomains=num_cores,numGPUs=num_GPUs,queue=None,scratch='',type=ANALYSIS,
    userSubroutine='',waitHours=0, waitMinutes=0)

# Submission ***********************************************************************************************
mdb.jobs[job_name].submit(consistencyChecking=OFF)
mdb.jobs[job_name].waitForCompletion()

# Elements Close to the Vertices ***************************************************************************
allElements1 = frame_instance.elements
allElements2 = skin_instance.elements
delta = 2e-2
g = 0; q = 0; frame_elements = []; skin_elements = [];
for i in range(0,len(p_array)):
    myElements1 = allElements1.getByBoundingSphere(center = p_array[i],radius=delta)
    for j in myElements1:
        frame_elements.append(j.label)
    myElements2 = allElements2.getByBoundingSphere(center = p_array[i],radius=delta)
    for j in myElements2:
        skin_elements.append(j.label)

# Writes the Elements #s close to the vertices
f = open('results_elements_in_vertices_frame.dat','w')
f.write('Element #s close to the vertices\n\n')
for j in frame_elements:
    f.write('%d\n'%(j))

f.close()

f = open('results_elements_in_vertices_skin.dat','w')
f.write('Element #s close to the vertices\n\n')
for j in skin_elements:
    f.write('%d\n'%(j))

f.close()
```

B.6 Output Extractor Example

The following extractor code pertains to the icosahedron model.

```
# -*- coding: mbcs -*-
import os
from part import*
from material import*
from section import*
from assembly import*
from step import*
from interaction import*
from load import*
from mesh import*
from job import*
from sketch import*
from visualization import*
from connectorBehavior import*
from odbAccess import *
from abaqusConstants import *
from odbMaterial import *
from odbSection import *
session.journalOptions.setValues(replayGeometry=COORDINATE, recoverGeometry=COORDINATE)

# Sets Working Directory ******************************************************************************
os.chdir(path)

# Load variables ***************************************************************************
execfile('Var_icosahedron_output.py')

# Extract Outputs ***************************************************************************************

# Opens the ODB and establishes the frame
odb=session.openOdb(name=job_name_odb, readOnly=False)
assembly = odb.rootAssembly

# Writes the Mesh Information for each Instance
f = open('results_meshdata.dat','w')
# Writes the element connectivity per instance
f.write('ELEMENT CONNECTIVITY\n')
f.write('Instance   Number   Type   Connectivity\n')
for name, instance in assembly.instances.items():
    for element in instance.elements:
        f.write('%s    '%(element.instanceName))
        f.write('%5d%8s' % (element.label, element.type),)
        for nodeNum in element.connectivity:
           f.write('  %4d' % nodeNum,)

        f.write('\n')
    f.write('\n')

f.close()

# Writes nodes coordinates for each instance in separate files
for name, instance in assembly.instances.items():
    f = open('results_' + str(name) +'_nodes_coordinates.dat','w')
```

```
    f.write('%s ************************************\n' % (name))
    if instance.embeddedSpace == THREE_D:
        f.write('#    X         Y         Z\n')
        for node in instance.nodes:
            f.write('%d  '%(node.label))
            f.write('%.5f  %.5f  %.5f\n'%(node.coordinates[0],node.coordinates[1],node.coordinates[2]))
    else:
        f.write('#    X         Y\n')
        for node in instance.nodes:
            f.write('%d  '%(node.label))
            f.write('%.5f  %.5f  %.5f\n'%(node.coordinates[0],node.coordinates[1],node.coordinates[2]))
    f.write('\n')
    f.close()


if buckle == 1:
    # Buckling Eigen Values
    f = open('buckling_eigen_values.dat','w')
    for num in odb.steps[stepname].frames:
        f.write('%s\n'%(num.description))
else:
    # Writes the Displacement for each instance for each Frame
    f = open('results_U.dat','w')
    f.write('LPF(or increment)          Instance          Node          U1          U2          U3\n')
    for frame in odb.steps[stepname].frames:
        lpf = frame.frameValue # load factor
        for node in frame.fieldOutputs['U'].values:
            f.write('%f '%(lpf))
            f.write('  %s '%(node.instance.name))
            f.write('  %d '%(node.nodeLabel))
            f.write('  %.12e  %.12e  %.12e\n'%(node.dataDouble[0],node.dataDouble[1],node.dataDouble[2]))

    f.close()

    # Writes the Nodal Forces for each instance for each Frame
    f = open('results_NF.dat','w')
    f.write('LPF(or increment)          Instance          Node          NF1
                NF2          NF3          NF4          NF5       NF6\n')
    for frame in odb.steps[stepname].frames:
        lpf  = frame.frameValue # load factor
        nf1  = frame.fieldOutputs['NFORC1'].values
        nf2  = frame.fieldOutputs['NFORC2'].values
        nf3  = frame.fieldOutputs['NFORC3'].values
        nf4  = frame.fieldOutputs['NFORC4'].values
        nf5  = frame.fieldOutputs['NFORC5'].values
        nf6  = frame.fieldOutputs['NFORC6'].values
        for node in xrange(len(nf1)):
            f.write('%f  '%(lpf))
            f.write('  %s  '%(nf1[node].instance.name))
            f.write('  %d  '%(nf1[node].nodeLabel))
            f.write('  %.12e  %.12e  %.12e  %.12e  %.12e  %.12e\n'%(nf1[node].data,nf2[node].data,
                      nf3[node].data,nf4[node].data,nf5[node].data,nf6[node].data))

    f.close()

    # Writes elements stress S1, S2, S3, Mises, Tresca for each instance for each Frame
    f = open('results_S.dat','w')
    f.write('LPF(or increment)          Instance          Element          S1          S2
```

```
                            S3          Mises          Spress\n')
    for frame in odb.steps[stepname].frames:
        lpf = frame.frameValue # load factor
        for element in frame.fieldOutputs['S'].values:
            f.write('%f   '%(lpf))
            f.write('  %s   '%(element.instance.name))
            f.write('  %d   '%(element.elementLabel))
            f.write('  %.3f  %.3f  %.3f  %.3f %.3f\n'%(element.data[0],element.data[1],element.data[2],
                        element.mises,element.press))

    f.close()

    # Writes the Strain Energy versus Time for the Whole Model
    f = open('results_SE.dat','w')
    energy = odb.steps[stepname].historyRegions['Assembly ASSEMBLY'].historyOutputs['ALLSE']
    f.write('Strain Energy versus Time for the whole model\n')
    f.write('Time    Strain Energy\n')
    for time in energy.data:
        f.write('%f    %.20f\n'%(time[0],time[1],))

    f.close()

odb.close()
```

B.7 Finite Elements Settings, Material Inputs and Geometric Inputs Example

The following code defines the FE settings, the material inputs and the geometric inputs that establish the icosahedron model.

```
p_array = [(0.000000,0.000000,0.152400),(0.136311,0.000000,0.068155),(0.042122,0.129639,0.068155),
(-0.110278,0.080121,0.068155),(-0.110278,-0.080121,0.068155),(0.042122,-0.129639,0.068155),
(0.110278,0.080121,-0.068155),(-0.042122,0.129639,-0.068155),(-0.136311,0.000000,-0.068155),
(-0.042122,-0.129639,-0.068155),(0.110278,-0.080121,-0.068155),(0.000000,0.000000,-0.152400),]

k = [(1,2,3),(1,3,4),(1,4,5),(1,5,6),(1,6,2),(12,7,8),(12,8,9),(12,9,10),(12,10,11),(12,11,7),(7,2,3),
(3,7,8),(8,3,4),(4,8,9),(9,4,5),(5,9,10),(10,5,6),(6,10,11),(11,2,6),(2,7,11),]

fc = [(0.059478,0.043213,0.096237),(-0.022718,0.069920,0.096237),(-0.073518,0.000000,0.096237),
(-0.022718,-0.069920,0.096237),(0.059478,-0.043213,0.096237),(0.022718,0.069920,-0.096237),
(-0.059478,0.043213,-0.096237),(-0.059478,-0.043213,-0.096237),(0.022718,-0.069920,-0.096237),
(0.073518,-0.000000,-0.096237),(0.096237,0.069920,0.022718),(0.036759,0.113133,-0.022718),
(-0.036759,0.113133,0.022718),(-0.096237,0.069920,-0.022718),(-0.118955,0.000000,0.022718),
(-0.096237,-0.069920,-0.022718),(-0.036759,-0.113133,0.022718),(0.036759,-0.113133,-0.022718),
(0.096237,-0.069920,0.022718),(0.118955,-0.000000,-0.022718),]

np = [(0.029739,0.021607,0.124318),(0.097894,0.021607,0.082196),(0.050800,0.086426,0.082196),
(-0.011359,0.034960,0.124318),(0.009702,0.099780,0.082196),(-0.066498,0.075021,0.082196),
(-0.036759,0.000000,0.124318),(-0.091898,0.040061,0.082196),(-0.091898,-0.040061,0.082196),
(-0.011359,-0.034960,0.124318),(-0.066498,-0.075021,0.082196),(0.009702,-0.099780,0.082196),
(0.029739,-0.021607,0.124318),(0.050800,-0.086426,0.082196),(0.097894,-0.021607,0.082196),
(0.011359,0.034960,-0.124318),(0.066498,0.075021,-0.082196),(-0.009702,0.099780,-0.082196),
(-0.029739,0.021607,-0.124318),(-0.050800,0.086426,-0.082196),(-0.097894,0.021607,-0.082196),
```

(-0.029739,-0.021607,-0.124318),(-0.097894,-0.021607,-0.082196),(-0.050800,-0.086426,-0.082196),
(0.011359,-0.034960,-0.124318),(-0.009702,-0.099780,-0.082196),(0.066498,-0.075021,-0.082196),
(0.036759,-0.000000,-0.124318),(0.091898,-0.040061,-0.082196),(0.091898,0.040061,-0.082196),
(0.103257,0.075021,-0.022718),(0.116274,0.034960,0.045437),(0.069180,0.099780,0.045437),
(0.039441,0.121386,0.022718),(0.073518,0.096627,-0.045437),(-0.002682,0.121386,-0.045437),
(-0.039441,0.121386,-0.022718),(0.002682,0.121386,0.045437),(-0.073518,0.096627,0.045437),
(-0.103257,0.075021,0.022718),(-0.069180,0.099780,-0.045437),(-0.116274,0.034960,-0.045437),
(-0.127633,0.000000,-0.022718),(-0.114617,0.040061,0.045437),(-0.114617,-0.040061,0.045437),
(-0.103257,-0.075021,0.022718),(-0.116274,-0.034960,-0.045437),(-0.069180,-0.099780,-0.045437),
(-0.039441,-0.121386,-0.022718),(-0.073518,-0.096627,0.045437),(0.002682,-0.121386,0.045437),
(0.039441,-0.121386,0.022718),(-0.002682,-0.121386,-0.045437),(0.073518,-0.096627,-0.045437),
(0.103257,-0.075021,-0.022718),(0.116274,-0.034960,0.045437),(0.069180,-0.099780,0.045437),
(0.127633,-0.000000,0.022718),(0.114617,0.040061,-0.045437),(0.114617,-0.040061,-0.045437),]

p1=(0.000000,0.000000,0.152400)
p2=(0.136311,0.000000,0.068155)
p3=(0.042122,0.129639,0.068155)
p4=(-0.110278,0.080121,0.068155)
p5=(-0.110278,-0.080121,0.068155)
p6=(0.042122,-0.129639,0.068155)
p7=(0.110278,0.080121,-0.068155)
p8=(-0.042122,0.129639,-0.068155)
p9=(-0.136311,0.000000,-0.068155)
p10=(-0.042122,-0.129639,-0.068155)
p11=(0.110278,-0.080121,-0.068155)
p12=(0.000000,0.000000,-0.152400)

mp12=(0.068155,0.000000,0.110278)
mp13=(0.021061,0.064820,0.110278)
mp23=(0.089217,0.064820,0.068155)
mp13=(0.021061,0.064820,0.110278)
mp14=(-0.055139,0.040061,0.110278)
mp34=(-0.034078,0.104880,0.068155)
mp14=(-0.055139,0.040061,0.110278)
mp15=(-0.055139,-0.040061,0.110278)
mp45=(-0.110278,0.000000,0.068155)
mp15=(-0.055139,-0.040061,0.110278)
mp16=(0.021061,-0.064820,0.110278)
mp56=(-0.034078,-0.104880,0.068155)
mp16=(0.021061,-0.064820,0.110278)
mp12=(0.068155,0.000000,0.110278)
mp62=(0.089217,-0.064820,0.068155)
mp127=(0.055139,0.040061,-0.110278)
mp128=(-0.021061,0.064820,-0.110278)
mp78=(0.034078,0.104880,-0.068155)
mp128=(-0.021061,0.064820,-0.110278)
mp129=(-0.068155,0.000000,-0.110278)
mp89=(-0.089217,0.064820,-0.068155)
mp129=(-0.068155,0.000000,-0.110278)
mp1210=(-0.021061,-0.064820,-0.110278)
mp910=(-0.089217,-0.064820,-0.068155)
mp1211=(0.055139,-0.040061,-0.110278)
mp1210=(-0.021061,-0.064820,-0.110278)
mp1110=(0.034078,-0.104880,-0.068155)
mp127=(0.055139,0.040061,-0.110278)
mp1211=(0.055139,-0.040061,-0.110278)
mp711=(0.110278,-0.000000,-0.068155)
mp211=(0.123294,-0.040061,0.000000)

```
mp27=(0.123294,0.040061,0.000000)
mp711=(0.110278,-0.000000,-0.068155)
mp112=(0.123294,-0.040061,0.000000)
mp116=(0.076200,-0.104880,0.000000)
mp26=(0.089217,-0.064820,0.068155)
mp610=(-0.000000,-0.129639,0.000000)
mp611=(0.076200,-0.104880,0.000000)
mp1011=(0.034078,-0.104880,-0.068155)
mp105=(-0.076200,-0.104880,0.000000)
mp106=(-0.000000,-0.129639,0.000000)
mp56=(-0.034078,-0.104880,0.068155)
mp59=(-0.123294,-0.040061,0.000000)
mp510=(-0.076200,-0.104880,0.000000)
mp910=(-0.089217,-0.064820,-0.068155)
mp94=(-0.123294,0.040061,0.000000)
mp95=(-0.123294,-0.040061,0.000000)
mp45=(-0.110278,0.000000,0.068155)
mp48=(-0.076200,0.104880,0.000000)
mp49=(-0.123294,0.040061,0.000000)
mp89=(-0.089217,0.064820,-0.068155)
mp83=(0.000000,0.129639,0.000000)
mp84=(-0.076200,0.104880,0.000000)
mp34=(-0.034078,0.104880,0.068155)
mp37=(0.076200,0.104880,0.000000)
mp38=(0.000000,0.129639,0.000000)
mp78=(0.034078,0.104880,-0.068155)
mp72=(0.123294,0.040061,0.000000)
mp73=(0.076200,0.104880,0.000000)
mp23=(0.089217,0.064820,0.068155)

fc123=(0.059478,0.043213,0.096237)
fc134=(-0.022718,0.069920,0.096237)
fc145=(-0.073518,0.000000,0.096237)
fc156=(-0.022718,-0.069920,0.096237)
fc162=(0.059478,-0.043213,0.096237)
fc1278=(0.022718,0.069920,-0.096237)
fc1289=(-0.059478,0.043213,-0.096237)
fc12910=(-0.059478,-0.043213,-0.096237)
fc121011=(0.022718,-0.069920,-0.096237)
fc12117=(0.073518,-0.000000,-0.096237)
fc723=(0.096237,0.069920,0.022718)
fc378=(0.036759,0.113133,-0.022718)
fc834=(-0.036759,0.113133,0.022718)
fc489=(-0.096237,0.069920,-0.022718)
fc945=(-0.118955,0.000000,0.022718)
fc5910=(-0.096237,-0.069920,-0.022718)
fc1056=(-0.036759,-0.113133,0.022718)
fc61011=(0.036759,-0.113133,-0.022718)
fc1126=(0.096237,-0.069920,0.022718)
fc2711=(0.118955,-0.000000,-0.022718)

disp_control = 0
P=101325.000000
d=-5.000000e-03

hollow_profile = 1
frame_beam_radius = 1.945217e-03
frame_beam_thickness = 9.726084e-05
```

```
frame_density = 970.000000
frame_poisson = 0.330000
frame_modulus = 1.720000e+11
frame_seed_number = 18
frame_element_type = B32

skin_thickness = 1.178366e-05
skin_density = 1650.000000
skin_poisson = 0.200000
skin_modulus = 1.000000e+12
skin_seed_number = 18
skin_element_type1 = M3D3
skin_element_type2 = M3D3
skin_element_shape = TRI

skin_section_idealization = NO_IDEALIZATION
skin_section_location = MIDDLE_SURFACE
membrane = 1
no_stiffness_skin = 0

rays_select = 0
hollow_profile_rays = 1
rays_beam_radius = 0.000000e+00
rays_beam_thickness = 0.000000e+00
rays_density = 1650.000000
rays_poisson = 0.200000
rays_modulus = 1.000000e+12
rays_seed_number = 18
rays_element_type = B32

stiff_select = 0
hollow_profile_stiff = 1
stiff_beam_radius = 0.000000e+00
stiff_beam_thickness = 0.000000e+00
stiff_density = 1650.000000
stiff_poisson = 0.200000
stiff_modulus = 1.000000e+12
stiff_seed_number = 18
stiff_element_type = B32

rotations = OFF

model_name = 'icosahedron-Model'
job_name = 'icosahedron-Job'
job_name_odb = 'icosahedron-Job.odb'

# Step Information
buckle = 0
stabilization = 1
step_type = 0
nonlinear_effects = ON
increment_method = AUTOMATIC
stepname = 'Nonlinear-Static,General-wStabi'

# If Buckle
buck_num_Eigen = 5
buck_max_Iter = 30
buck_num_vectors = 30
```

```
# If General
initial_inc = 1.000000e+00
max_inc = 1.000000e+00
min_inc = 1.000000e-36
stabilization_ratio = 0.0500000000
stabilization_magn = 0.0002000000

# If Riks
initial_ArcInc = 1.000000e-01
min_ArcInc = 1.000000e-12
max_ArcInc = 1.000000e+00
maxnuminc = 100000000
max_LPF = 2.000000e+00

num_cores = 1
memory_usage = 1024
num_GPUs = 0
```

Appendix C: Matlab Codes

The Matlab codes shown here are related to the icosahedron model, but similar codes where used to analysis the other models.

C.1 Main Routine: Main.m

```
% delete('*.inp','*.com','*.log','*.ipm','*.sim','*.msg',...
%        '*.rec','*.rpy','*.dat','*.sta','*.prt','*.lck','*.log'); clear f
clc; clear all; close all
%% Optimization Rutine
% Last updated: Jan 17, 2014
% ************************************************************************
%% Input
I.scratch_folder = 'Temp Scratch Files'; % used to create the scratch folder and the enviroment .env file
% Job Info (Parallel Processing, memory allocation, use of GPUs)
I.job.num_cores = 1; % # of cores used in the analysis
I.job.memory_usage = 1024; %  amount of allocated memory, MB
I.job.num_GPUs = 0; % number of GPUs (graphics processing units) used, 0 for none

% Static Step Info
I.step.buckle = 0; % ON(1) / OFF(0), ON disables others
I.step.stabilization = 1; % stabilization ON(1) / OFF(0), ON w/membrane section, ON diables Riks
I.step.step_type = 0; % use Riks(1), use General(0); use General(0) w/membrane section
I.step.nonlinear_effects = 'ON'; % ON or OFF, ON w/membrane section
I.step.increment_method = 'AUTOMATIC'; % Increments (arc length if Riks) method: 'FIXED' or 'AUTOMATIC'
I.step.maxnuminc = 100000000; % max number of increments, if fixed
    % Static Riks
    I.step.initial_ArcInc = 0.1; % initial arc length
    I.step.min_ArcInc = 1e-12; % minimum arc length
    I.step.max_ArcInc = 1; % maximum arc length
    I.step.max_LPF = 2; % max load proportionality factor
    % Static General
    I.step.initial_inc = 1e-3; % starting time increment
    I.step.max_inc = 1; % max time increment
    I.step.min_inc = 1e-36; % min time increment
    I.step.stabilization_ratio = 0.05; % w/membrane only - adaptive stabilization:
                                       % max stabilization/strain energy ratio, default = 0.05
    I.step.stabilization_magn = 0.0002; % w/membrane only - dissipated energy fraction, default = 0.0002
    % Linear Buckle
    I.step.buck_num_Eigen   = 5;
    I.step.buck_max_Iter    = 30;
    I.step.buck_num_vectors = 30;

% Load
I.load.disp_control = 1; % displacement(1), load(0) controls
I.load.d = -3e2 ; % m, displacement control BC
I.load.P = 101325; % Pa, sea level pressure (safety factor 1.5)

% Skin Sections
```

```
I.section.no_stiffness_skin = 0; % 0(no) or 1(yes); rigid skin, use surface elements
I.section.membrane          = 1; % membrane section (1), shell section (0)
    % Shell Only
    I.section.skin_section_idealization = 'NO_IDEALIZATION'; % 'MEMBRANE','BENDING','NO_IDEALIZATION'
    I.section.skin_section_location     = 'MIDDLE_SURFACE'; % 'MIDDLE_SURFACE','TOP_SURFACE','BOTTOM_SURFACE'

% Tie Constraint
I.tie.rotations = 'OFF'; % tie rotations between skin/frame

% Mesh
I.mesh.skin_element_type1 = 'M3D3'; % See 'Shell and Membrane Element Library Info.txt'
I.mesh.skin_element_type2 = 'M3D3';
I.mesh.skin_element_shape = 'TRI'; % Element shape: rectangular or triangular
I.mesh.skin_seed_number   =  30  ; % skin # of elements/edge, 30 edges in total
I.mesh.frame_element_type = 'B32'; % need to use beam element type: B31, B32, etc.
I.mesh.frame_seed_number  =  30  ; % frame # of elements/edge, 30 edges in total
I.mesh.rays_element_type  = 'B32'; % need to use beam element type: B31, B32, etc.
I.mesh.rays_seed_number   =  18  ; % rays # of elements/edge, 20 edges in total
I.mesh.stiff_element_type = 'B32'; % need to use beam element type: B31, B32, etc.
I.mesh.stiff_seed_number  =  18  ; % rays # of elements/edge, 60 edges in total

% Parameters for W/B ratio calculation
I.W_B.rho   = 1.2754; % air density at SL, kg/m^3, http://en.wikipedia.org/wiki/Density_of_air
I.W_B.g     = 9.81; % acceleration of gravity, m/s^2
I.W_B.skin  = 0.4;  % skin W/B ratio set value
I.W_B.frame = 0.5; % frame W/B ratio set value
I.W_B.rays  = 0; % rays W/B ratio set value
I.W_B.stiff = 0; % rays W/B ratio set value
I.W_B.To    = 293.15; % K, external temp (altitude dependent)
I.W_B.Ti    = To; % K, internal temp (altitude and heat transfer dependent)
I.W_B.Po    = 101325; % Pa, external pressure (altitude dependent)

%% Geometry and Material Selection
% Material Selection
      %  rho   nu     E      Sy    ; % Units: kg/m^3,-,Pa,Pa
mat1 = [1870  0.3   440e9   3.73e9 ]; % UHM Unidirectional Carbon Epoxy tubes
mat2 = [1560  0.37  303e9   5.8e9  ]; % Zylon
mat3 = [2700  0.12  757e9   75.7e9 ]; % Diamond like Carbon, or Diamond thin film,
                     % yield aprox Y = E/10: see p1795,'Paper - Diamond like Carbon' in references
mat4 = [2570  0.33  400e9   3.6e9  ]; % Boron Monofilament, nu guessed
mat5 = [1650  0.2   1000e9  10e9   ]; % Nanocyl NANOCYL¢ NC7000 Thin Multi-Wall Carbon Nanotubes
                       % , nu aprox: see 'Paper - Study of Poisson Ratios of Graphene and Nanotubes'
mat6 = [1844  0.18  303e9   0.4e9  ]; % Beryllium S-200, Tubing
mat7 = [2650  0.18  379e9   1.7e9  ]; % CoorsTek Boron Carbide Reaction-Bonded Boron Carbide
mat8 = [2800  0.33  738e9   0.14e9 ]; % Vista Metals Duramold-2¢ Cast Aluminum Mold Plate, nu aprox
mat9 = [247   0.33  5.76e9  0.024e9]; % 3A Composites Core Materials BALTEK
                     % SB.150 Structural End-Grain Balsa, nu aprox
mat10= [970   0.33  172e9   3.0e9  ]; % Honeywell Spectra 1000 Fiber

% Material Assignment
matf = mat10; % assigned frame material (from the selection above)
mats = mat10; % assigned skin  material (from the selection above)
matr = mat10; % assigned rays  material (from the selection above)
matst= mat10; % assigned stiffners  material (from the selection above)

I.materials.frame_density = matf(1); I.materials.frame_poisson   = matf(2);
I.materials.frame_modulus = matf(3); I.materials.frame_yield     = matf(4);
I.materials.skin_density  = mats(1); I.materials.skin_poisson    = mats(2);
```

```
I.materials.skin_modulus  = mats(3); I.materials.skin_yield      = mats(4);
I.materials.rays_density  = matr(1); I.materials.rays_poisson    = matr(2);
I.materials.rays_modulus  = matr(3); I.materials.rays_yield      = matr(4);
I.materials.stiff_density = matst(1); I.materials.stiff_poisson  = matst(2);
I.materials.stiff_modulus = matst(3); I.materials.stiff_yield    = matst(4);

% Geometry (icosahedron)
I.geometry.r = 0.1524; % icosahedron radius, m; 0.1524 m = 1/2 ft
I.geometry.rays = 0; % rays off(0), rays on(1)
I.geometry.stiff = 0; % rays off(0), rays on(1)
I.section.hollow_profile_rays = 1; % Rays beam profile: hollow(1),solid(0); beam t ignored if (0)
I.section.hollow_profile_stiff= 1; % Stiff beam profile: hollow(1),solid(0); beam t ignored if (0)
I.section.hollow_profile      = 1; % Frame beam profile: hollow(1),solid(0); beam t ignored if (0)
c1 = 0.05; % frame     : if hollow circular beam, t = c*r_beam
c2 = 0.05; % rays      : if hollow circular beam, t = c*r_beam
c3 = 0.05; % stiffners : if hollow circular beam, t = c*r_beam

for i = 1
I.geometry.skin_thickness = I.geometry.r*I.W_B.rho*I.W_B.skin/...
                            (3.77523*I.materials.skin_density); % m, ensures the WB skin set value
if I.section.hollow_profile == 1
    I.geometry.frame_beam_radius    = I.geometry.r*sqrt(I.W_B.frame*I.W_B.rho/...
                                      (39.0742*(2*c1-c1^2)*I.materials.frame_density)); % meters
    I.geometry.frame_beam_thickness = c1*I.geometry.frame_beam_radius; % input
else
    I.geometry.frame_beam_radius    = sqrt(I.geometry.r^2*I.W_B.rho*I.W_B.frame/...
                                           (39.0742*I.materials.frame_density)); % m,
    I.geometry.frame_beam_thickness = I.geometry.frame_beam_radius;
end
if I.section.hollow_profile_rays == 1
    I.geometry.rays_beam_radius    = I.geometry.r*sqrt(I.W_B.rays*I.W_B.rho/...
                                     (19.82*(2*c2-c2^2)*I.materials.rays_density)); % meters
    I.geometry.rays_beam_thickness = c2*I.geometry.rays_beam_radius; % input
else
    I.geometry.rays_beam_radius    = sqrt(I.geometry.r^2*I.W_B.rho*I.W_B.rays/...
                                           (19.82*I.materials.rays_density)); % m,
    I.geometry.rays_beam_thickness = I.geometry.rays_beam_radius;
end
if I.section.hollow_profile_stiff == 1
    I.geometry.stiff_beam_radius    = I.geometry.r*sqrt(I.W_B.stiff*I.W_B.rho/...
                                      (45.34*(2*c3-c3^2)*I.materials.stiff_density)); % meters
    I.geometry.stiff_beam_thickness = c3*I.geometry.stiff_beam_radius; % input
else
    I.geometry.stiff_beam_radius    = sqrt(I.geometry.r^2*I.W_B.rho*I.W_B.stiff/...
                                           (45.34*I.materials.stiff_density)); % m,
    I.geometry.stiff_beam_thickness = I.geometry.stiff_beam_radius;
end
```

C.2 Icosahedron Coordinates Function: icosahedron_coordinates.m

```
% Outputs the icosahedron vertices cartesian xyz coordinates given the
% radius (r) and frequency(f). f =1 is an 20 triangle/30 edges Icosahedron
% Coding cortesy of Capt Trent Metlen, created on 10/10/2012
% Function created on 08/09/2013
```

```
function [XYZ]=icosahedron_coordinates(r,f)

% Angles in spherical coordinates used to determine icosahedron vertices
phi1=0.4636476090008;
phi2=pi/2-phi1;
theta1=2*pi/5;

% Icosahedron verticies in spherical coordinates
Vp=[0 pi/2;0 phi1;theta1 phi1;2*theta1 phi1;3*theta1 phi1;4*theta1 phi1;
       0.5*theta1 -phi1;1.5*theta1 -phi1;2.5*theta1 -phi1;3.5*theta1 -phi1; 4.5*theta1 -phi1;0 -pi/2];

% Basic information on geodesic shape
nV=10*f^2+2; %number of vertices
nt=20*f^2;  %number of triangles
ne=30*f^2; %number of edges

% Find all vertices of first 5 major triangles for geoesic sphere
V=[r*ones(nV,1) ones(nV,2)];
V(1,2:3)=Vp(1,:);
V(nV,2:3)=Vp(12,:);
K=1;
 for I=1:f
    for J=1:I*5
        V((K+J),2:3)=[(J-1)*2*pi/(I*5) pi/2-I*phi2/f];
    end
    J;
    K=K+J;
end

% Find all vertices of next 10 major triangles
for I=1:f
    for J=1:5*f
        V((K+J),2:3)=[I*(pi/(5*f))+(2*pi/(5*f))*(J-1) phi1-2*phi1*I/f];
    end
    J;
    K=K+J;
end

% Find all vertices of last 5 major triangles
for I=1:f
    if I<f
        for J=1:5*(f-I)
V((K+J),2:3)=[pi/5+(2*pi/(5*(f-I)))*(J-1) -pi/2+phi2-phi2*I/f];
        end
        K=K+J;
    end
end

% Convert from spherical coordinates to cartesian coordinates
[x y z]=sph2cart(V(:,2)',V(:,3)',V(:,1)');
leg=sqrt((x(1)-x(2))^2+(y(1)-y(2))^2+(z(1)-z(2))^2);

% Create xyz matrix in cartesian coordinates
XYZ=[x' y' z'];
```

C.3 Icosahedron Caller Function 1: icosahedron_fea_inner.m

```
% By Adorno-Rodriguez, Ruben
% Last updated: Jan 15, 2014
% Function: runs the icosahedorn FEA model
function [output]=icosahedron_fea_inner(I)
%% Input
% Filename
filename = 'icosahedron'; % I.filename; % .py filename
model_name = [filename,'-Model'];
job_name = [filename,'-Job'];
job_name_odb = [filename,'-Job.odb'];
scratch_folder = I.scratch_folder; % used to create the scratch folder and the enviroment .env file

% Job Info (Parallel Processing, memory allocation, use of GPUs)
num_cores = I.job.num_cores; % # of cores used in the analysis
memory_usage = I.job.memory_usage; %  amount of allocated memory, MB
num_GPUs = I.job.num_GPUs; % number of GPUs (graphics processing units) used, 0 for none

% Static Step Information
increment_method = I.step.increment_method; % Increments (arc length if Riks) method: 'FIXED' or 'AUTOMATIC'
nonlinear_effects = I.step.nonlinear_effects; % ON or OFF
buckle = I.step.buckle; % ON(1) / OFF(0), ON disables others
step_type = I.step.step_type; % use Riks(1), use General(0)
stabilization = I.step.stabilization; % strain energy stabilization ON(1) / OFF(0), ON w/membrane section
    % Static Riks
    initial_ArcInc = I.step.initial_ArcInc; % initial arc length
    min_ArcInc = I.step.min_ArcInc; % minimum arc length
    max_ArcInc = I.step.max_ArcInc; % maximum arc length
    max_LPF = I.step.max_LPF; % max load proportionality factor
    % Static General
    initial_inc = I.step.initial_inc; % starting time increment
    max_inc = I.step.max_inc; % max time increment
    min_inc = I.step.min_inc; % min time increment
    maxnuminc = I.step.maxnuminc; % max number of increments
    stabilization_ratio = I.step.stabilization_ratio;
    stabilization_magn = I.step.stabilization_magn;
    % Buckle
    buck_num_Eigen   = I.step.buck_num_Eigen;
    buck_max_Iter    = I.step.buck_max_Iter;
    buck_num_vectors = I.step.buck_num_vectors;
if buckle == 1
    stepname = 'Buckle';
elseif strcmp(nonlinear_effects,'ON') && step_type == 1
    stepname = 'Nonlinear-Riks';
    stabilization = 0;
elseif strcmp(nonlinear_effects,'ON') && step_type == 0 && stabilization == 1
    stepname = 'Nonlinear-Static,General-wStabi';
elseif strcmp(nonlinear_effects,'ON') && step_type == 0 && stabilization == 0
    stepname = 'Nonlinear-Static,General';
elseif strcmp(nonlinear_effects,'OFF') && step_type == 1
    stepname = 'Linear-Riks';
    stabilization = 0;
else
    stepname = 'Linear-Static,General';
end
```

```
% Geometry
r = I.geometry.r; % icosahedron radius, meters
skin_thickness = I.geometry.skin_thickness; % meters
frame_beam_radius = I.geometry.frame_beam_radius ; % meters
frame_beam_thickness = I.geometry.frame_beam_thickness; % meters
rays_select = I.geometry.rays; % rays off(0), rays on(1)
rays_beam_radius = I.geometry.rays_beam_radius ; % meters
rays_beam_thickness = I.geometry.rays_beam_thickness; % meters
stiff_select = I.geometry.stiff; % stiffners off(0), rays on(1)
stiff_beam_radius = I.geometry.stiff_beam_radius ; % meters
stiff_beam_thickness = I.geometry.stiff_beam_thickness; % meters

% Material
frame_density = I.materials.frame_density; % kg/m^3
frame_poisson = I.materials.frame_poisson;
frame_modulus = I.materials.frame_modulus; % Pa
frame_yield   = I.materials.frame_yield; %Pa
skin_density  = I.materials.skin_density; % kg/m^3
skin_poisson  = I.materials.skin_poisson;
skin_modulus  = I.materials.skin_modulus; % Pa
skin_yield    = I.materials.skin_yield; %Pa
rays_density  = I.materials.rays_density; % kg/m^3
rays_poisson  = I.materials.rays_poisson;
rays_modulus  = I.materials.rays_modulus; % Pa
rays_yield    = I.materials.rays_yield; %Pa
stiff_density  = I.materials.stiff_density; % kg/m^3
stiff_poisson  = I.materials.stiff_poisson;
stiff_modulus  = I.materials.stiff_modulus; % Pa
stiff_yield    = I.materials.stiff_yield; %Pa

% Load
disp_control = I.load.disp_control; % displacement(1), load(0) controls
d = I.load.d; % m, displacement control BC
P = I.load.P; % Pa, sea level pressure


% Frame Profile
hollow_profile = I.section.hollow_profile; % hollow(1),solid(0); beam thickness ignored if (0)
hollow_profile_rays = I.section.hollow_profile_rays; % hollow(1),solid(0); beam thickness ignored if (0)
hollow_profile_stiff = I.section.hollow_profile_stiff; % hollow(1),solid(0); beam thickness ignored if (0)

% Skin Sections
no_stiffness_skin = I.section.no_stiffness_skin;% 0(no) or 1(yes); rigid skin, use surface elements
membrane = I.section.membrane; % membrane section (1), shell section (0)
skin_section_idealization = I.section.skin_section_idealization; % MEMBRANE, BENDING or NO_IDEALIZATION
skin_section_location = I.section.skin_section_location; % 'MIDDLE_SURFACE', 'TOP_SURFACE' or 'BOTTOM_SURFACE'

% Tie Constraint
rotations = I.tie.rotations; % tie rotations between skin/frame

% Mesh
skin_seed_number   = I.mesh.skin_seed_number  ; % size seeding
skin_element_type1 = I.mesh.skin_element_type1; % See 'Shell and Membrane Element Library Info.txt'
skin_element_type2 = I.mesh.skin_element_type2;
skin_element_shape = I.mesh.skin_element_shape; % Element shape: rectangular or triangular
frame_element_type = I.mesh.frame_element_type; % need to use beam element type: B31, B32, etc.
frame_seed_number  = I.mesh.frame_seed_number ; % # of elements/edge, 30 edges in total
rays_element_type  = I.mesh.rays_element_type; % need to use beam element type: B31, B32, etc.
```

```
rays_seed_number   = I.mesh.rays_seed_number ; % # of elements/edge, 20 edges in total
stiff_element_type  = I.mesh.stiff_element_type; % need to use beam element type: B31, B32, etc.
stiff_seed_number   = I.mesh.stiff_seed_number ; % # of elements/edge, 60 edges in total

%% Geometry Calculations
% Calculates the icosahedron vertices
p=icosahedron_coordinates(r,1);
% Calculates the vertices in cartesian coordinates
c=[1 1 2,1 1 3,1 1 4,1 1 5,1 1 6,12 12 7,12 12 8,12 12 9 ,12 12 11,12 12 7 ,...
   2  2 7 ,11 11 2,6  6  10,10 10 5,5 5  9 ,9 9 4,4 4 8,8 8 3,3 3 7,7 7 2;
   2 3 3,3 4 4,4 5 5,5 6 6,6 2 2,7  8  8,8  9  9,9  10 10,11 10 10,7  11 11,...
   11 7 11,2  6  6,10 11 11,5  6  6,9 10 10,4 5 5,8 9 9,3 4 4,7 8 8,2 3 3]; %connectivity array
t=0.5; % used to select midpoint
for i=1:length(c) %calculates the midpoint
    mp(i,:)=[p(c(1,i),1)+t*(p(c(2,i),1)-p(c(1,i),1)),...
             p(c(1,i),2)+t*(p(c(2,i),2)-p(c(1,i),2)),...
             p(c(1,i),3)+t*(p(c(2,i),3)-p(c(1,i),3))];
end
%Calculates center of each face
k=[1 1 1 1 1 12 12 12 12 12 7 3 8 4 9 5  10 6  11 2 ;
   2 3 4 5 6 7  8  9  10 11 2 7 3 8 4 9  5  10 2  7 ;
   3 4 5 6 2 8  9  10 11 7  3 8 4 9 5 10 6  11 6  11];
for i=1:length(k)
    fc(i,:)=[mean([p(k(1,i),1) p(k(2,i),1) p(k(3,i),1)])...
             mean([p(k(1,i),2) p(k(2,i),2) p(k(3,i),2)])...
             mean([p(k(1,i),3) p(k(2,i),3) p(k(3,i),3)])];
end
% Calculates center between each face and each vertex
g = 1;
for j=1:length(k)
    for i = 1:3
    np(g,:)=[p(k(i,j),1)+t*(fc(j,1)-p(k(i,j),1)),...
             p(k(i,j),2)+t*(fc(j,2)-p(k(i,j),2)),...
             p(k(i,j),3)+t*(fc(j,3)-p(k(i,j),3))];
         g = g + 1;
    end
end
%Calculates the distance of each edge (to confirm coordinates accuracy)
for i=1:length(c)
    edge_length(i)=sqrt((p(c(2,i),1)-p(c(1,i),1))^2+...
                        (p(c(2,i),2)-p(c(1,i),2))^2+...
                        (p(c(2,i),3)-p(c(1,i),3))^2);
end

%% Writes variables into Var.py file,which will be read by the main .py file
fid = fopen(['Var_',filename,'.py'],'w');
% Arrays
fprintf(fid,'p_array = ['); % writes vertices in a array
for i=1:length(p)
    fprintf(fid,'(%0.6f,%0.6f,%0.6f),',p(i,1),p(i,2),p(i,3));
end
fprintf(fid,']');
fprintf(fid,'\r\n\r\n');

fprintf(fid,'k = ['); % writes faces connectivity array
for i=1:length(k)
    fprintf(fid,'(%d,%d,%d),',k(1,i),k(2,i),k(3,i));
end
```

```
fprintf(fid,']');
fprintf(fid,'\r\n\r\n');

fprintf(fid,'fc = ['); % writes face center array
for i=1:length(k)
    fprintf(fid,'(%0.6f,%0.6f,%0.6f),',fc(i,1),fc(i,2),fc(i,3));
end
fprintf(fid,']');
fprintf(fid,'\r\n\r\n');

fprintf(fid,'np = ['); % writes center between each face and each vertex
for i=1:length(np(:,1))
    fprintf(fid,'(%0.6f,%0.6f,%0.6f),',np(i,1),np(i,2),np(i,3));
end
fprintf(fid,']');
fprintf(fid,'\r\n\r\n');

for i=1:length(p) %writes the vertices
    fprintf(fid,'p%d=(%0.6f,%0.6f,%0.6f)\r\n',i,p(i,1),p(i,2),p(i,3));
end
fprintf(fid,'\r\n');
for i=1:length(c) % writes the midpoints
    fprintf(fid,'mp%d%d=(%0.6f,%0.6f,%0.6f)\r\n',c(1,i),c(2,i),mp(i,1),mp(i,2),mp(i,3));
end
fprintf(fid,'\r\n');
for i=1:length(k) % writes the faces centers
    fprintf(fid,'fc%d%d%d=(%0.6f,%0.6f,%0.6f)\r\n',k(1,i),k(2,i),k(3,i),fc(i,1),fc(i,2),fc(i,3));
end
fprintf(fid,'\r\n');
fprintf(fid,'disp_control = %d\r\n',disp_control);
fprintf(fid,'P=%0.6f\r\n',P); % writes the pressure
fprintf(fid,'d=%0.6e',d); % writes the displacement control BC
fprintf(fid,'\r\n\r\n');
% Frame
fprintf(fid,'hollow_profile = %d\r\n',hollow_profile);
fprintf(fid,'frame_beam_radius = %0.6e\r\n',frame_beam_radius);
fprintf(fid,'frame_beam_thickness = %0.6e\r\n',frame_beam_thickness);
fprintf(fid,'frame_density = %0.6f\r\n',frame_density);
fprintf(fid,'frame_poisson = %0.6f\r\n',frame_poisson);
fprintf(fid,'frame_modulus = %0.6e\r\n',frame_modulus);
fprintf(fid,'frame_seed_number = %d\r\n',frame_seed_number);
fprintf(fid,'frame_element_type = %s\r\n \r\n',frame_element_type);
% Skin
fprintf(fid,'skin_thickness = %0.6e\r\n',skin_thickness);
fprintf(fid,'skin_density = %0.6f\r\n',skin_density);
fprintf(fid,'skin_poisson = %0.6f\r\n',skin_poisson);
fprintf(fid,'skin_modulus = %0.6e\r\n',skin_modulus);
fprintf(fid,'skin_seed_number = %d\r\n',skin_seed_number);
fprintf(fid,'skin_element_type1 = %s\r\n',skin_element_type1);
fprintf(fid,'skin_element_type2 = %s\r\n',skin_element_type2);
fprintf(fid,'skin_element_shape = %s\r\n \r\n',skin_element_shape);
fprintf(fid,'skin_section_idealization = %s\r\n',skin_section_idealization);
fprintf(fid,'skin_section_location = %s\r\n',skin_section_location);
fprintf(fid,'membrane = %d\r\n',membrane);
fprintf(fid,'no_stiffness_skin = %d\r\n \r\n',no_stiffness_skin);
% Rays
fprintf(fid,'rays_select = %d\r\n',rays_select);
fprintf(fid,'hollow_profile_rays = %d\r\n',hollow_profile_rays);
```

```
fprintf(fid,'rays_beam_radius = %0.6e\r\n',rays_beam_radius);
fprintf(fid,'rays_beam_thickness = %0.6e\r\n',rays_beam_thickness);
fprintf(fid,'rays_density = %0.6f\r\n',rays_density);
fprintf(fid,'rays_poisson = %0.6f\r\n',rays_poisson);
fprintf(fid,'rays_modulus = %0.6e\r\n',rays_modulus);
fprintf(fid,'rays_seed_number = %d\r\n',rays_seed_number);
fprintf(fid,'rays_element_type = %s\r\n \r\n',rays_element_type);
% Stiffners
fprintf(fid,'stiff_select = %d\r\n',stiff_select);
fprintf(fid,'hollow_profile_stiff = %d\r\n',hollow_profile_stiff);
fprintf(fid,'stiff_beam_radius = %0.6e\r\n',stiff_beam_radius);
fprintf(fid,'stiff_beam_thickness = %0.6e\r\n',stiff_beam_thickness);
fprintf(fid,'stiff_density = %0.6f\r\n',stiff_density);
fprintf(fid,'stiff_poisson = %0.6f\r\n',stiff_poisson);
fprintf(fid,'stiff_modulus = %0.6e\r\n',stiff_modulus);
fprintf(fid,'stiff_seed_number = %d\r\n',stiff_seed_number);
fprintf(fid,'stiff_element_type = %s\r\n \r\n',stiff_element_type);
% Tie Constraint
fprintf(fid,'rotations = %s\r\n \r\n',rotations);
% Model names
fprintf(fid,'model_name = ''%s''\r\n',model_name);
fprintf(fid,'job_name = ''%s''\r\n',job_name);
fprintf(fid,'job_name_odb = ''%s''\r\n \r\n',job_name_odb);
% Step Information
fprintf(fid,'# Step Information\r\n');
fprintf(fid,'buckle = %d\r\n',buckle);
fprintf(fid,'stabilization = %d\r\n',stabilization);
fprintf(fid,'step_type = %d\r\n',step_type);
fprintf(fid,'nonlinear_effects = %s\r\n',nonlinear_effects);
fprintf(fid,'increment_method = %s\r\n',increment_method);
fprintf(fid,'stepname = ''%s''\r\n',stepname);
fprintf(fid,'\r\n# If Buckle\r\n');
fprintf(fid,'buck_num_Eigen = %d\r\n',buck_num_Eigen);
fprintf(fid,'buck_max_Iter = %d\r\n',buck_max_Iter);
fprintf(fid,'buck_num_vectors = %d\r\n',buck_num_vectors);
fprintf(fid,'\r\n# If General\r\n');
fprintf(fid,'initial_inc = %e\r\n',initial_inc);
fprintf(fid,'max_inc = %e\r\n',max_inc);
fprintf(fid,'min_inc = %e\r\n',min_inc);
fprintf(fid,'stabilization_ratio = %0.10f\r\n',stabilization_ratio);
fprintf(fid,'stabilization_magn = %0.10f\r\n',stabilization_magn);
fprintf(fid,'\r\n# If Riks\r\n');
fprintf(fid,'initial_ArcInc = %e\r\n',initial_ArcInc);
fprintf(fid,'min_ArcInc = %e\r\n',min_ArcInc);
fprintf(fid,'max_ArcInc = %e\r\n',max_ArcInc);
fprintf(fid,'maxnuminc = %d\r\n',maxnuminc);
fprintf(fid,'max_LPF = %e\r\n \r\n',max_LPF);
% Job information
fprintf(fid,'num_cores = %d\r\n',num_cores);
fprintf(fid,'memory_usage = %d\r\n',memory_usage);
fprintf(fid,'num_GPUs = %d\r\n',num_GPUs);
% Add the path to the python code to be evaluated
f = fopen([filename,'.py'],'r'); A = fread(f); fclose(f);
f = fopen(['python2abaqus_',filename,'.py'],'w');
fprintf(f,'path = ''%s''\r\n \r\n',pwd); % writes the current directory
fwrite(f,A); fclose(f);
% Writes the environment file w/the scratch folder's directory
f = fopen('environment.env','r'); A = fread(f); fclose(f);
```

```
f = fopen('abaqus_v6.env','w');
fprintf(f,['scratch=''%s\\',scratch_folder,'''\r\n \r\n'],pwd); % writes the scratch directory
fwrite(f,A); fclose(f);

%% Runs the Adjusted Script in Abaqus
% warning('off','all');
% delete('*.inp','*.com','*.log','*.ipm','*.sim','*.msg',...
%        '*.rec','*.rpy','*.dat','*.sta','*.prt','*.lck','*.log'); clear f
% warning('on','all');
Rmo = 'noGUI'; % No GUI, analysis runs in the background
mo = 'script'; % W/GUI, analysis runs with Abaqus GUI
[status,cmdout] = system(['abaqus cae ',Rmo,'=python2abaqus_',filename,'.py']); % runs the main script

%% Output
output.system.status = status; % 0 if succesful, nonzero otherwise
output.system.cmdout = cmdout; % detailed message
output.geometry.vertices    = p;  % vertices
output.geometry.midpoints   = mp; % edge midpoints
output.geometry.facecenters = fc; % face centers
end
```

C.4 Icosahedron Caller Function 2: icosahedron_fea.m

```
% By Adorno-Rodriguez, Ruben
% Last updated: Jan 15, 2014
% Function: runs the FEA model of the icosahedron in Abaqus and reads in
% results
function [output]=icosahedron_fea(I)
%% Input
% Static Step Information
step_type = I.step.step_type; % use Riks(1), use General(0)

%% Runs the icosahedron_fea_inner(I) function
% Runs the FEA Analysis
O1 = icosahedron_fea_inner(I);
status = O1.system.status; % 0 if succesful, nonzero otherwise
cmdout = O1.system.cmdout; % detailed message
% If an error occurs, it changes the initial increment on the Newton
% Raphson Step for convergence
if status == 0 % 0(no error),otherwise(error)
    disp('Analysis completed succesfully!')
else
    arc_length = [1e-1 1e-2 1e-3 1e-4 1e-5 1e-6 1e-8]; % arc length sweep for Riks
    increment =  [1e-1 1e-2 1e-3 1e-4 1e-5 1e-6 1e-8]; % increment  sweep for Newthon Raphson
    j = 1; % initializes counter
    while status ~= 0 % 0(no error),otherwise(error)
        disp('Abaqus Message:'); disp(cmdout);
        if step_type == 0 % Riks(1), Newton Raphson General(0)
            fprintf('Step failed, initial increment changed to %.1e.\n',increment(j))
            I.step.initial_inc = increment(j); % starting time increment
        else
            fprintf('Step failed, initial arc length changed to %.1e.\n',arc_length(j))
            I.step.initial_ArcInc = arc_length(j);
        end
        fprintf('Analysis reinitialized\n')
```

```
        O1 = icosahedron_fea_inner(I);
        status = O1.system.status; % 0 if succesful, nonzero otherwise
        cmdout = O1.system.cmdout; % detailed message
        if j == length(increment) || j == length(arc_length) % stops while loop at last entry
            disp('All the initial increments in the increment vector failed. Function will stop')
            break
        end
        j = j + 1;
        if status == 0 % 0(no error),otherwise(error)
            disp('Analysis completed succesfully!')
        end
    end
end

%% Reads-in and Saves the FEA outputs
% Geometry
output.geometry.vertices    = O1.geometry.vertices;  % vertices
output.geometry.midpoints   = O1.geometry.midpoints; % edge midpoints
output.geometry.facecenters = O1.geometry.facecenters; % face centers

% Nodes coordinates
    % Frame Instance
    f = fopen('results_Frame Instance_nodes_coordinates.dat');
    coordinates = textscan(f,'%.1f %.5f %.5f %.5f','HeaderLines',2);
    fclose(f);
    coordinates_frame = cell2mat(coordinates); % node #, x, y, z
    % Skin Instance
    f = fopen('results_Skin Instance_nodes_coordinates.dat');
    coordinates = textscan(f,'%.1f %.5f %.5f %.5f','HeaderLines',2);
    fclose(f);
    coordinates_skin = cell2mat(coordinates); % node #, x, y, z

% Mesh Details
f = fopen('results_meshdata.dat');
mesh = textscan(f,'%s %*s %f %s %f %f %f %*[^\n]','HeaderLines',2);
fclose(f);
s  = cell2mat(mesh(:,2)); % element #
s1 = s(strcmpi(mesh{:,1},{'Frame'}) == 1,:);
s2 = s(strcmpi(mesh{:,1},{'Skin'}) == 1,:);
output.mesh.nodes    = coordinates_frame(end,1) + coordinates_skin(end,1);
output.mesh.elements = max(s1) + max(s2);

% System
output.system.status = status; % 0 if succesful, nonzero otherwise
output.system.cmdout = cmdout; % detailed message

% Eigen values
if I.step.buckle == 1
f = fopen('buckling_eigen_values.dat','r');
A = textscan(f,'%s %s %s %s %f','HeaderLines',1);
fclose(f);
output.buckling = [[1:length(A{end})]' A{end}]; % frame #,eigen value:Pcrit = Po*eigen(i)
else

% Nodes Displacements
f = fopen('results_U.dat');
displacement = textscan(f,'%f %s %*s %f %f %f %f %*[^\n]','HeaderLines',1);
fclose(f);
```

```
inc = unique(displacement{:,1}); % increment (General) or LPF (Riks)
output.inc(:,1) = inc;
d  = cell2mat(displacement(:,[1 3:end])); % increment, node #, U1, U2, U3
d1 = d(strcmpi(displacement{:,2},{'Frame'}) == 1,:);
d2 = d(strcmpi(displacement{:,2},{'Skin'}) == 1,:);
for i = 1:length(inc)
    output.frame(i,1).U = [coordinates_frame d1(inc(i) == d1(:,1),3:end)]; % node #, x, y, z, U1, U2, U3
    output.skin (i,1).U = [coordinates_skin  d2(inc(i) == d2(:,1),3:end)]; % node #, x, y, z, U1, U2, U3
end

% Nodal Forces
f = fopen('results_NF.dat');
nodalforces = textscan(f,'%f %s %*s %f %f %f %f %f %f %f %*[^\n]','HeaderLines',1);
fclose(f);
nf  = cell2mat(nodalforces(:,[1 3:end])); % increment, node #, U1, U2, U3
nf1 = nf(strcmpi(nodalforces{:,2},{'Frame'}) == 1,:);
nf2 = nf(strcmpi(nodalforces{:,2},{'Skin'}) == 1,:);
for i = 1:length(inc)
    nf1i = nf1(inc(i) == nf1(:,1),2:end);
    nf2i = nf2(inc(i) == nf2(:,1),2:end);
    for j = 1:max(nf1i(:,1))
        k = find(nf1i(:,1) == j);
        if length(k) == 1
            nf1ij(j,:) = nf1i(k,:);
        else
            nf1ij(j,:) = mean(nf1i(k,:));
        end
    end
    for j = 1:max(nf2i(:,1))
        k = find(nf2i(:,1) == j);
        if length(k) == 1
            nf2ij(j,:) = nf2i(k,:);
        else
            nf2ij(j,:) = mean(nf2i(k,:));
        end
    end
    output.frame(i,1).NF = [coordinates_frame nf1ij(:,2:end)]; % node #, x, y, z, NF1, NF2, NF3, NF4, NF5, NF6
    output.skin (i,1).NF = [coordinates_skin  nf2ij(:,2:end)]; % node #, x, y, z, NF1, NF2, NF3, NF4, NF5, NF6
end

% Elements Stresses
f = fopen('results_S.dat');
stress = textscan(f,'%f %s %*s %f %f %f %f %f %*[^\n]','HeaderLines',1);
fclose(f);
s  = cell2mat(stress(:,[1 3:end])); % increment, element #, S1, S2, S3, Mises, Spressure
s1 = s(strcmpi(stress{:,2},{'Frame'}) == 1,:);
s2 = s(strcmpi(stress{:,2},{'Skin'}) == 1,:);
for i = 1:length(inc)
    output.frame(i,1).S = s1(inc(i) == s1(:,1),2:end); % element #, S1, S2, S3, Mises, Spress
    output.skin (i,1).S = s2(inc(i) == s2(:,1),2:end); % element #, S1, S2, S3, Mises, Spress
end

% Strain Energy versus Time
f = fopen('results_SE.dat');
strain_energy = textscan(f,'%f %.20f %*[^\n]','HeaderLines',2);
fclose(f);
output.strain_energy = cell2mat(strain_energy); % increment or load factor, strain energy
end
```

```
%% Output Structure Summary
% output
%     .system
%         .status: 0 if succesful, nonzero otherwise
%         .cmdout: detailed message
%     .geometry
%         .vertices  : icosahedron vertices
%         .midpoints : edge midpoints
%         .facecenters: face centers
%     .mesh
%         nodes   : total # of nodes
%         elements: total # of elements
%     if buckle == 1
%        .buckling: frame #,eigen value:Pcrit = Po*eigen(i)
%     else
%       .inc: column of increments or load factors
%       .frame/skin: for each increment or load factor(i),
%            .U : node #, x, y, z, U1, U2, U3 (displacement)
%            .NF: node #, x, y, z, NF1, NF2, NF3, NF4, NF5, NF6 (nodal forces)
%            .S : elem #, S1, S2, S3, Mises, Spressure (stresses)
%       .strain_energy: increment(s) or load factor, strain energy(J)
end
```

C.5 Icosahedron Importer Function 1: icosahedron_fea_output1.m

```
% By Adorno-Rodriguez, Ruben
% Last updated: Jan 15, 2014
% Function: reads the results from the odb file and creates .dat files
function [output]=icosahedron_fea_output1(I)
%% Input
% Filename
filename2 = 'icosahedron_output';
filename = 'icosahedron'; % I.filename; % .py filename
model_name = [filename,'-Model'];
job_name = [filename,'-Job'];
job_name_odb = [filename,'-Job.odb'];

% Static Step Information
increment_method = I.step.increment_method; % Increments (arc length if Riks) method: 'FIXED' or 'AUTOMATIC'
nonlinear_effects = I.step.nonlinear_effects; % ON or OFF
buckle = I.step.buckle; % ON(1) / OFF(0), ON disables others
step_type = I.step.step_type; % use Riks(1), use General(0)
stabilization = I.step.stabilization; % strain energy stabilization ON(1) / OFF(0), ON w/membrane section
if buckle == 1
    stepname = 'Buckle';
elseif strcmp(nonlinear_effects,'ON') && step_type == 1
    stepname = 'Nonlinear-Riks';
    stabilization = 0;
elseif strcmp(nonlinear_effects,'ON') && step_type == 0 && stabilization == 1
    stepname = 'Nonlinear-Static,General-wStabi';
elseif strcmp(nonlinear_effects,'ON') && step_type == 0 && stabilization == 0
    stepname = 'Nonlinear-Static,General';
elseif strcmp(nonlinear_effects,'OFF') && step_type == 1
```

```
    stepname = 'Linear-Riks';
    stabilization = 0;
else
    stepname = 'Linear-Static,General';
end

%% Writes variables into Var.py file,which will be read by the main .py file
fid = fopen(['Var_',filename2,'.py'],'w');
% Model names
fprintf(fid,'job_name_odb = ''%s''\r\n \r\n',job_name_odb);
% Step Information
fprintf(fid,'# Step Information\r\n');
fprintf(fid,'buckle = %d\r\n',buckle);
fprintf(fid,'stabilization = %d\r\n',stabilization);
fprintf(fid,'step_type = %d\r\n',step_type);
fprintf(fid,'stepname = ''%s''\r\n',stepname);
fprintf(fid,'nonlinear_effects = %s\r\n',nonlinear_effects);
fclose(fid);

% Add the path to the python code to be evaluated
f = fopen([filename2,'.py'],'r'); A = fread(f); fclose(f);
f = fopen(['python2abaqus_',filename2,'.py'],'w');
fprintf(f,'path = ''%s''\r\n \r\n',pwd); % writes the current directory
fwrite(f,A); fclose(f);

%% Runs the Adjusted Script in Abaqus
Rmo = 'noGUI'; % No GUI, analysis runs in the background
mo = 'script'; % W/GUI, analysis runs with Abaqus GUI
[status,cmdout] = system(['abaqus cae ',Rmo,'=python2abaqus_',filename2,'.py']); % runs the main script
delete('*.rpy','*.lck','*.log');
delete(['python2abaqus_',filename2,'.py']);
delete(['Var_',filename2,'.py']);

%% Output
output.status = status;
output.cmdout = cmdout;
end
```

C.6 Icosahedron Coordinates Function2: icosahedron_output2.m

```
% By Adorno-Rodriguez, Ruben
% Last updated: Jan 15, 2014
% Function: reads in the results from the .dat files
function [output]=icosahedron_fea_output2(I)
%% Input
% Geometry
r = I.geometry.r; % icosahedron radius, meters
% Static Step Information
step_type = I.step.step_type; % use Riks(1), use General(0)

%% Calculates Geometry
% Calculates the icosahedron vertices
p=icosahedron_coordinates(r,1);

% Calculates the vertices in cartesian coordinates
```

```
c=[1 1 2,1 1 3,1 1 4,1 1 5,1 1 6,12 12 7,12 12 8,12 12 9 ,12 12 11,12 12 7 ,...
   2  2 7 ,11 11 2,6  6  10,10 10 5,5 5  9 ,9 9 4,4 4 8,8 8 3,3 3 7,7 7 2;
   2 3 3,3 4 4,4 5 5,5 6 6,6 2 2,7  8  8,8  9  9,9  10 10,11 10 10,7  11 11,...
   11 7 11,2  6  6,10 11 11,5  6  6,9 10 10,4 5 5,8 9 9,3 4 4,7 8 8,2 3 3]; %connectivity array
t=0.5; % used to select midpoint
for i=1:length(c) %calculates the midpoint
    mp(i,:)=[p(c(1,i),1)+t*(p(c(2,i),1)-p(c(1,i),1)),...
             p(c(1,i),2)+t*(p(c(2,i),2)-p(c(1,i),2)),...
             p(c(1,i),3)+t*(p(c(2,i),3)-p(c(1,i),3))];
end
%Calculates center of each face
k=[1 1 1 1 1 12 12 12 12 12 7 3 8 4 9 5  10 6  11 2 ;
   2 3 4 5 6 7  8  9  10 11 2 7 3 8 4 9  5  10 2  7 ;
   3 4 5 6 2 8  9  10 11 7  3 8 4 9 5 10 6  11 6  11];
for i=1:length(k)
    fc(i,:)=[mean([p(k(1,i),1) p(k(2,i),1) p(k(3,i),1)])...
             mean([p(k(1,i),2) p(k(2,i),2) p(k(3,i),2)])...
             mean([p(k(1,i),3) p(k(2,i),3) p(k(3,i),3)])];
end


%% Reads-in and Saves the FEA outputs
% Geometry
output.geometry.vertices    = p;  % vertices
output.geometry.midpoints   = mp; % edge midpoints
output.geometry.facecenters = fc; % face centers

% Nodes coordinates
    % Frame Instance
    f = fopen('results_Frame Instance_nodes_coordinates.dat');
    coordinates = textscan(f,'%.1f %.5f %.5f %.5f','HeaderLines',2);
    fclose(f);
    coordinates_frame = cell2mat(coordinates); % node #, x, y, z
    % Skin Instance
    f = fopen('results_Skin Instance_nodes_coordinates.dat');
    coordinates = textscan(f,'%.1f %.5f %.5f %.5f','HeaderLines',2);
    fclose(f);
    coordinates_skin = cell2mat(coordinates); % node #, x, y, z

% Mesh Details
f = fopen('results_meshdata.dat');
mesh = textscan(f,'%s %*s %f %s %f %f %f %*[^\n]','HeaderLines',2);
fclose(f);
s  = cell2mat(mesh(:,2)); % element #
e1 = s(strcmpi(mesh{:,1},{'Frame'}) == 1,:);
e2 = s(strcmpi(mesh{:,1},{'Skin'}) == 1,:);
output.mesh.nodes    = coordinates_frame(end,1) + coordinates_skin(end,1);
output.mesh.elements = max(e1) + max(e2);

% Eigen values
if I.step.buckle == 1
f = fopen('buckling_eigen_values.dat','r');
A = textscan(f,'%s %s %s %s %f','HeaderLines',1);
fclose(f);
output.buckling = [[1:length(A{end})]' A{end}]; % frame #,eigen value:Pcrit = Po*eigen(i)
else

% Nodes Displacements
```

```
f = fopen('results_U.dat');
displacement = textscan(f,'%f %s %*s %f %f %f %f %*[^\n]','HeaderLines',1);
fclose(f);
inc = unique(displacement{:,1}); % increment (General) or LPF (Riks)
output.inc(:,1) = inc;
d  = cell2mat(displacement(:,[1 3:end])); % increment, node #, U1, U2, U3
d1 = d(strcmpi(displacement{:,2},{'Frame'}) == 1,:);
d2 = d(strcmpi(displacement{:,2},{'Skin'}) == 1,:);
for i = 1:length(inc)
    a1 = d1(inc(i) == d1(:,1),2:end);
    for j = 1:coordinates_frame(end,1)
        a2(j,:) = [mean(a1(a1(:,1) == j,2)), mean(a1(a1(:,1) == j,3)),...
                   mean(a1(a1(:,1) == j,4))];
    end
    b1 = d2(inc(i) == d2(:,1),2:end);
    for j = 1:coordinates_skin(end,1)
        b2(j,:) = [mean(b1(b1(:,1) == j,2)), mean(b1(b1(:,1) == j,3)),...
                   mean(b1(b1(:,1) == j,4))];
    end
    output.frame(i,1).U = [coordinates_frame a2]; % node #, x, y, z, U1, U2, U3
    output.skin (i,1).U = [coordinates_skin  b2]; % node #, x, y, z, U1, U2, U3
end

% Nodal Forces
f = fopen('results_NF.dat');
nodalforces = textscan(f,'%f %s %*s %f %f %f %f %f %f %f %*[^\n]','HeaderLines',1);
fclose(f);
nf  = cell2mat(nodalforces(:,[1 3:end])); % increment, node #, U1, U2, U3
nf1 = nf(strcmpi(nodalforces{:,2},{'Frame'}) == 1,:);
nf2 = nf(strcmpi(nodalforces{:,2},{'Skin'}) == 1,:);
for i = 1:length(inc)
    nf1i = nf1(inc(i) == nf1(:,1),2:end);
    nf2i = nf2(inc(i) == nf2(:,1),2:end);
    for j = 1:max(nf1i(:,1))
        k = find(nf1i(:,1) == j);
        if length(k) == 1
            nf1ij(j,:) = nf1i(k,:);
        else
            nf1ij(j,:) = mean(nf1i(k,:));
        end
    end
    for j = 1:max(nf2i(:,1))
        k = find(nf2i(:,1) == j);
        if length(k) == 1
            nf2ij(j,:) = nf2i(k,:);
        else
            nf2ij(j,:) = mean(nf2i(k,:));
        end
    end
    output.frame(i,1).NF = [coordinates_frame nf1ij(:,2:end)]; % node #, x, y, z, NF1, NF2, NF3, NF4, NF5, NF6
    output.skin (i,1).NF = [coordinates_skin  nf2ij(:,2:end)]; % node #, x, y, z, NF1, NF2, NF3, NF4, NF5, NF6
end

% Elements Stresses
f = fopen('results_S.dat');
stress = textscan(f,'%f %s %*s %f %f %f %f %f %f %*[^\n]','HeaderLines',1);
fclose(f);
s  = cell2mat(stress(:,[1 3:end])); % increment, element #, S1, S2, S3, Mises, Spressure
```

```
s1 = s(strcmpi(stress{:,2},{'Frame'}) == 1,:);
s2 = s(strcmpi(stress{:,2},{'Skin'}) == 1,:);
for i = 1:length(inc)
    output.frame(i,1).S = s1(inc(i) == s1(:,1),2:end); % element #, S1, S2, S3, Mises, Spress
    output.skin (i,1).S = s2(inc(i) == s2(:,1),2:end); % element #, S1, S2, S3, Mises, Spress
end

% Strain Energy versus Time
f = fopen('results_SE.dat');
strain_energy = textscan(f,'%f %.20f %*[^\n]','HeaderLines',2);
fclose(f);
output.strain_energy = cell2mat(strain_energy); % increment or load factor, strain energy

% W/B including Volume Reduction
To = I.W_B.To; % K, external temp (altitude dependent)
Ti = I.W_B.Ti; % K, internal temp (altitude and heat transfer dependent)
Po = I.W_B.Po; % Pa, external pressure (altitude dependent)
R  = I.W_B.R ; % J/(kg-K), air specific gas constant
for i = 1:length(inc)
    x = output.skin(i).U(:,2);
    y = output.skin(i).U(:,3);
    z = output.skin(i).U(:,4);
    U1 = output.skin(i).U(:,5);
    U2 = output.skin(i).U(:,6);
    U3 = output.skin(i).U(:,7);
    igeom = [x y z]; % coordinates of the initial geometry
    fgeom = [x + U1, y + U2, z + U3]; % coordinates of the final geometry
    iDT = delaunayTriangulation(igeom);
    [iK,iv] = convexHull(iDT); % initial volume was checked againts theory eq.: good!
    fDT = delaunayTriangulation(fgeom);
    [fK,fv] = convexHull(fDT);
    Vframe = 30*pi*(2*I.geometry.frame_beam_radius*I.geometry.frame_beam_thickness-...
                    I.geometry.frame_beam_thickness^2)*(I.geometry.r/5*sqrt(50-10*sqrt(5))); % frame volume
    Vr = iv - fv - Vframe; % volume reduction
    Pi = Po - inc(i)*Po; % Pa
    output.WB(i) = ((9.5745*I.geometry.skin_thickness*I.geometry.r^2*I.materials.skin_density)+...
                    (99.098*(2*I.geometry.frame_beam_radius*I.geometry.frame_beam_thickness-...
                    I.geometry.frame_beam_thickness^2)*I.geometry.r*I.materials.frame_density))/...
                    ((2.5362*I.geometry.r^3-Vr)*Po/(R*To))+Pi/Po*To/Ti; % final W/B icosahedron
     output.graph(i).iDT = iDT;% initial triangulation
     output.graph(i).iK  = iK; % initial points
     output.graph(i).iv  = iv; % initial volume
     output.graph(i).fDT = fDT;% final triangulation
     output.graph(i).fK  = fK; % final points
     output.graph(i).fv  = fv; % final volume
     output.Vr(i)        = iv - Vr; % volume reduction
end

end

%% Output Structure Summary
% output
%     .system
%         .status: 0 if succesful, nonzero otherwise
%         .cmdout: detailed message
%     .geometry
%         .vertices  : icosahedron vertices
%         .midpoints : edge midpoints
```

```
%         .facecenters: face centers
%     .mesh
%         nodes   : total # of nodes
%         elements: total # of elements
%     if buckle == 1
%        .buckling: frame #,eigen value:Pcrit = Po*eigen(i)
%     else
%       .inc: column of increments or load factors
%       .frame/skin: for each increment or load factor(i),
%            .U : node #, x, y, z, U1, U2, U3 (displacement)
%            .NF: node #, x, y, z, NF1, NF2, NF3, NF4, NF5, NF6 (nodal forces)
%            .S : elem #, S1, S2, S3, Mises, Spressure (stresses)
%       .strain_energy: increment(s) or load factor, strain energy(J)
%       .WB: W/B ratio
%       .Vr: volume reduction
%       .graph(i).iDT: initial triangulation
%                 .iK : initial points
%                 .iv : initial volume
%                 .fDT: final triangulation
%                 .fK : final points
%                 .fv : final volume
end
```

Bibliography

[1] *U.S. Standard Atmosphere 1976*. U.S. Government Printing Office, Washington, D.C., 1976.

[2] Akhmeteli, A. and A.V. Gavrilin. *Layered Shell Vacuum Balloons*. Patent application, May 2005. URL http://akhmeteli.org/lighter-than-air-solid.

[3] Ardema, Mark D. *Missions and Vehicle Concepts for Modern, Propelled, Lighter-than-Air Vehicles*. Report no.724, Advisory Group for Aerospace Research and Development, February 1985.

[4] Ardema, Mark D. *Understanding NManualar Analysis*. Report, Dassault Systemes SolidWorks Corp., 2008.

[5] Arnold, David. *Spherical Coordinates in Matlab*, February 2014. URL http://msemac.redwoods.edu/~darnold/math50c/mathjax/spherical/index.xhtml.

[6] Ashby, Michael. *Materials Selection in Mechanical Design*. Oxford, OX, 3rd edition, 1999. ISBN 0-7506-4357-9.

[7] Automation Creations Inc., Christiansburg, VA. *Matweb: the Manual Materials Information Resource*, 1996 - 2014. URL www.matweb.com.

[8] Barlow, J. "Optimal Stress Locations in Finite Element Models". volume 10, 243–251. 1976.

[9] CADTuto Blog. *Icosahedron?*, January 2013. URL http://www.cadtutor.net/forum/showthread.php?76524-Icosahedron.

[10] Century of Flight. *Jules Henri Giffard (1825 - 1882)*, January 2014. URL http://www.century-of-flight.net/Aviation%20history/to%20reality/Jules%20Henri%20Giffard.htm.

[11] Dassault Syste'mes, Providence, Rhode Island, 20010. *Abaqus v.6.11.1*.

[12] Dassault Syste'mes. *Abaqus Analysis User's Manual*, 2011.

[13] Dassault Syste'mes. *Abaqus Theory Manual*, 2011.

[14] Davis, Tom. *Lighter Than Air*. The Johns Hopkins University Press, 2009.

[15] Degenhardt, R., H. Klein, A. Kling, H. Temmen, and R. Zimmermann. *Buckling and Post Buckling Analysis of Shells under Quasi Static and Dynamic Loadings*. Unpublished article. URL http://www.dlr.de/fa/Portaldata/17/Resources/dokumente/institut/srw_05.pdf. Lilienthalplatz 7, 38108 Braunschweig.

[16] Driver, Bob. "P-791 Airship". *Los Angeles Times*, August 2012. URL http://www.latimes.com/business/lat-bcpix_m9l3empd20120830161257,0,624192.photo#axzz2sqJD4dIr.

[17] Fitzpatrick, Richard. *The Adiabatic Atmosphere*, January 2014. URL http://farside.ph.utexas.edu/teaching/sm1/lectures/node56.html.

[18] Geo-Dome. *Build a folded paper icosahedron*, January 2014. URL http://www.geodome.co.uk/article.asp?uname=paper_icosahedron.

[19] Geometry in Action. *Delaunay Triangulation*, February 2014. URL http://www.ics.uci.edu/~eppstein/gina/delaunay.html.

[20] Grossman, Dan. *Airships: The Hindenburg and other Zeppelins*, January 2014. URL http://www.airships.net/dirigible.

[21] Grossman, Dan. *Hydrogen Airship Disasters*, January 2014. URL http://www.airships.net/hydrogen-airship-accidents.

[22] Halliday, David, Robert Resnick, and Jearl Walker. *Fundamentals of Physics*. John Wiley and Sons, 2004. ISBN 978-0471216438.

[23] Hibbeler, R.C. *Mechanics of Materials*. Pearson Prentice Hall, 7th edition, 2008. ISBN 978-0-13-220991-5.

[24] Indiemaps Blog. *Delaunay Triangulation in ActionScript 3*, February 2014. URL http://indiemaps.com/blog/2008/05/delaunay-triangulation-in-actionscript-3/.

[25] Inforscience. *Andrews Convex Hull Algorithm*, February 2014. URL http://rendon.x10.mx/?p=159.

[26] Joseph, MacDonnell S.J. *Francesco Lana-Terzi, S.J. (1631-1687); The Father of Aeronautics*, January 2014. URL http://www.faculty.fairfield.edu/jmac/sj/scientists/lana.htm.

[27] Kurowski, Paul M. *Buckling Analysis with FEA*. Machine Design, February 2014. URL http://machinedesign.com/fea-and-simulation/buckling-analysis-fea.

[28] Lana-Terzi, Francesco. *The Aerial Ship*. The Aeronautical Society of Great Britain, 1910.

[29] Literka. *How to Compute Volume of a Regular 20-faces Polyhedron (Icosahedron)*, January 2014. URL http://www.literka.addr.com/ico_volume.htm.

[30] Livermore Software Technology: LS- DYNA. *Hourglass*, February 2014. URL http://www.dynasupport.com/howtos/element/hourglass.

[31] Math Open Reference. *Icosahedron*, January 2014. URL http://www.mathopenref.com/icosahedron.html.

[32] MATLAB. *Convex Hull*. Natick, Massachusetts, 2013. URL http://www.mathworks.com/help/matlab/ref/convhull.html.

[33] MATLAB. *Delaunay Triangulation Class*. The MathWorks Inc., Natick, Massachusetts, 2013. URL http://www.mathworks.com/help/matlab/ref/delaunaytriangulationclass.html.

[34] MATLAB. *version 8.1.0 (R2013a)*. Natick, Massachusetts, 2013.

[35] MatWeb. *Copper, Cu; Cold-Worked*, January 2014. URL http://matweb.com/search/DataSheet.aspx?MatGUID=ca486cc7cefa44d98ee67d2f5eb7d21f.

[36] MatWeb. *Nanocyl NC7000 Thin Multi-Wall Carbon Nanotubes*, January 2014. URL http://matweb.com/search/DataSheet.aspx?MatGUID=94228642b4eb40d882658fb7853f240b.

[37] MCS Software. *Finite Elements - Multi-Point Constraints*, February 2014. URL http://www.mscsoftware.com/training_videos/patran/Reverb_help/index.html#page/Marc/marc02_model.3.4.html.

[38] Metlen, Trent T. *Design of a Lighter than Air Vehicle that Achieves Positive Buoyancy in Air using a Vacuum*. Thesis, Air Force Institute of Technology, June 2005.

[39] Nautic Expo. *Sandwich panel : fiberglass / aluminium honeycomb*, June 2014. URL http://www.nauticexpo.com/prod/cel-components-srl/sandwich-panels-fiberglasss-aluminium-honeycombs-34324-244391.html.

[40] Orbital Technologies Corp. *Vacuum Reference Chart*, February 2014. URL http://www.orbitec.com/documents/Orbitec_Vacuum_Reference.pdf.

[41] Oxford Dictionaries. *Definition of Model*, June 2014. URL http://www.oxforddictionaries.com/us/definition/american_english/model.

[42] Palazotto, Anthony N. and Scott T. Dennis. *NManualar Analysis of Shell Structures*. American Institute of Aeronautics and Astronautics, 1992. ISBN 1-56347-033-0.

[43] Robertson, J. "Diamond-like Carbon". volume 66, 1795. 1994. URL http://pac.iupac.org/publications/pac/pdf/1994/pdf/6609x1789.pdf.

[44] van Rossum, Guido. *Python Programming Language*, 1990-2014.

[45] S. Timoshenko, S. Woinowsky-Krieger. *Theory of Plates and Shells*. McGraw-Hill, 2nd edition, 1959. ISBN 0-07-064779-8.

[46] Saada, Adel S. *Elasticity: Theory and Applications*. J. Ross Publishing, Inc., 2nd edition, 2009. ISBN 978-1-60427-019-8.

[47] Seide, Paul. "Large Deflections of Rectangular Membranes Under Uniform Pressure". volume 12, 397–406. 1977.

[48] Shampine, L.F. "Matlab program for quadrature in 2D". *Applied Mathematics and Computation*, 202(1):266 – 274, 2008. ISSN 0096-3003. URL http://www.sciencedirect.com/science/article/pii/S0096300308000982.

[49] Ugural, A.C. *Stresses in Plates and Shells*. McGraw-Hill, 1981. ISBN 0-07-065730-0.

[50] US Centennial of Flight Commission. *The Atmosphere*, February 2014. URL http://www.centennialofflight.net/essay/Theories_of_Flight/atmosphere/TH1.htm.

[51] Wolfram MathWorld. *Icosahedron*, January 2014. URL http://mathworld.wolfram.com/Icosahedron.html.

[52] Wolfram MathWorld. *Platonic Solid*, January 2014. URL http://mathworld.wolfram.com/PlatonicSolid.html.

[53] Wolfram MathWorld. *Regular Polygon*, January 2014. URL http://mathworld.wolfram.com/RegularPolygon.html.

[54] Wolfram MathWorld. *Regular Polyhedron*, January 2014. URL http://mathworld.wolfram.com/RegularPolyhedron.html.

[55] Wolfram MathWorld. *Triangulation*, February 2014. URL http://mathworld.wolfram.com/Triangulation.html.

[56] Zhao, P. and G. Shi. "Study of Poisson's Ratios of Graphene and Single-Walled Carbon Nanotubes Based on an Improved Molecular Structural Mechanics Model". volume 5, 49–58. 2011. URL http://pac.iupac.org/publications/pac/pdf/1994/pdf/6609x1789.pdf.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
27–03–2014	Master's Thesis	Oct 2012-Mar 2014

4. TITLE AND SUBTITLE

Nonlinear Structural Analysis of an Icosahedron and its Application to Lighter than Air Vehicles under a Vacuum

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Adorno-Rodriguez, Ruben, Second Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENY-14-M-03

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. David Stargel
AFOSR/RTA
875 N. Randolph Street
Suite 325, Room 3112
Arlington VA, 22203-1768
(703) 696-6961 (DSN 426-)

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The concept that a structure is capable of producing buoyancy using an internal vacuum rather than a gas dates back to the 1600s; but material technology has restricted the construction of such concept for common geometries, such as the sphere. Different and often complex geometries compensate for the lack of light materials that provide the stiffness and strength needed. Therefore, this research looks at an Lighter than Air Vehicle (LTAV) in the form of an icosahedral frame/skin configuration using nonlinear finite element analysis in order to determine the structural response of such vehicle, its capacity to sustain a vacuum with both material technology that exists today and in the near future, and its buoyancy characteristics. The structural response is characterized with large displacements; where membrane behavior dominates the icosahedral skin response, generating geometric stiffening in the overall structure. It is shown that those displacements have minimal effect in the structures buoyancy, with no more than 4% reduction. Overall, the nonlinear analysis of the icosahedral structure provided tangible background on its behavior and the LTAV applicability. It is feasibly possible to actually manufacture this type of vehicle in the very near future depending upon newer materials with more advanced strength.

15. SUBJECT TERMS

nonlinear, lighter than air vehicles, structural analysis, finite element analysis, membrane analysis, icosahedron

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Anthony Palazotto (ENY)
U	U	U	UU	233	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x4599 anthony.palazotto@afit.edu